

**PENGEMBANGAN SISTEM PENGELOLAAN
PROSES BISNIS CERDAS**

SKRIPSI

**Oleh :
FIKRI FIUCA FARDANA
NIM. 15650056**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK
IBRAHIM MALANG
2021**

**PENGEMBANGAN SISTEM PENGELOLAAN
PROSES BISNIS CERDAS**

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
FIKRI FIUCA FARDANA
NIM. 15650056**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN

**PENGEMBANGAN SISTEM PENGELOLAAN
PROSES BISNIS CERDAS**

SKRIPSI

Oleh :

**FIKRI FIUCA FARDANA
NIM.15650056**

Telah diperiksa dan disetujui untuk Diuji
Tanggal:

Pembimbing I

Pembimbing II

Muhammad Ainul Yaqin, M.Kom
NIP. 19761013 200604 1 004

Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr.Cahyo Crysdian
NIP.19740424 200901 1 008

LEMBAR PENGESAHAN
PENGEMBANGAN SISTEM PENGELOLAAN
PROSES BISNIS CERDAS

SKRIPSI

Oleh:

FIKRI FIUCA FARDANA
NIM. 15650056

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal 30 Desember 2020

Susunan Dewan Penguji		Tanda Tangan
1. Penguji Utama	: <u>Fajar Rohman Hariri M.Kom</u> NIP. 19890515 201801 1 001	()
2. Ketua Penguji	: <u>Ajib Hanani, M.T</u> NIDT. 19840731 20160801 1 076	()
3. Sekretaris Penguji	: <u>Muhammad Ainul Yaqin, M.Kom</u> NIP. 19761013 200604 1 004	()
4. Anggota Penguji	: <u>Syahiduz Zaman M.Kom</u> NIP. 19700502 200501 1 005	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Fikri Fiuca Fardana

NIM : 15650056

Fakultas/Jurusan : Sains dan Teknologi/Teknik Informatika

Judul Skripsi : Pengembangan Sistem Pengelolaan Bisnis Cerdas

Menyatakan dengan benar bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pusaka. Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Desember 2020

Yang Membuat Pernyataan



Fikri Fiuca Fardana

NIM. 15650056

HALAMAN MOTTO

*“Berikan yang terbaik pada usahamu
Berdoalah sungguh-sungguh untuk usahamu
Bersyukurlah atas hasil perih kerja usahamu
Entah apapun itu”*

HALAMAN PERSEMBAHAN

Alhamdulillah hirobbil Alamin atas kehadiran Allah SWT. Skripsi ini saya persembahkan untuk Bapak Choirul Anam selaku ayah penulis dan Ibu Tutik Supriati selaku ibu penulis yang telah mengisi dunia saya dengan begitu banyak kebahagiaan sehingga seumur hidup tidak cukup untuk menikmati semuanya. Terima kasih atas semua cinta yang telah ayah dan ibu berikan kepada saya.

Terimakasih pula saya ucapkan untuk pembimbing yang telah membimbing dalam melakukan penelitian ini dan memberikan motivasi serta dorongan hingga penelitian ini dapat terselesaikan dengan lancar.

Tidak lupa terima kasih saya ucapkan kepada teman-teman satu perjuangan jurusan Teknik Informatika 2015 UIN Maulana Malik Ibrahim Malang yang telah menemani dan mengisi hari-hari selama lima tahun terakhir. Dan kepada teman-teman yang selalu memberikan semangat dan bantuan dalam mengerjakan penelitian ini. Terima kasih untuk orang-orang yang tidak dapat disebutkan satu per satu yang telah memberikan motivasi, semangat dan doa sehingga penelitian skripsi ini dapat rampung dengan lancar.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr. Wb.

Dengan memanjatkan puja dan puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul “**Pengembangan Sistem Pengelolaan Proses Binsis Cerdas**”, sebagai salah satu syarat untuk menyelesaikan Program Sarjana (S1) Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Shalawat serta salam senantiasa selalu terlimpahkan kepada junjungan kita Nabi Muhammada SAW, keluarga dan para sahabat yang telah membimbing umat dari zaman jahiliyah menuju jalan yang diridhoi Allah SWT.

Penulis menyadari bahwa skripsi ini tidak mungkin terselesaikan tanpa adanya dukungan dan, bantuan, bimbingan, dan nasehat dari berbagai pihak selama penyusunan skripsi ini. Pada kesempatan ini penulis menyampaikan terima kasih setulus-tulusnya kepada :

1. Prof Dr H Abd. Haris, M.Ag selaku rektor UIN Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

4. Bapak Muhammad Ainul Yaqin selaku pembimbing I dan Syahiduz Zaman M.Kom selaku pembimbing II atas segala bimbingan, arahan, serta saran yang diberikan kepada penulis sehingga skripsi ini dapat terselasaikan dengan baik.
5. Seluruh Dosen Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang yang telah memberikan ilmu dan pengetahuan serta pengalaman yang sangat berharga dan bermanfaat.
6. Segenap civitas akademik Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
7. Kedua orang tua serta seluruh keluarga besar penulis yang selalu dengan senantiasa mendukung dan mendoakan penulis.
8. Sahabar-sahabat seperjuangan Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

Penulis berharap penelitian ini dapat bermanfaat kedepannya nanti. Penulis menyadari dalam karya ini masih banyak kekurangan. Oleh karena itu segala kritik dan saran dari pembaca sangat membantu dan dapat menjadi pertimbangan sehingga bisa lebih baik setelahnya. Semoga karya ini dapat bermanfaat bagi seluruh pihak.

Malang, 29 Desember 2020

Penulis

DAFTAR ISI

PENGEMBANGAN SISTEM PENGELOLAAN.....	i
PROSES BISNIS CERDAS	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TULISAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii
ABSTRAK.....	xiii
ABSTRACT	xiv
الملخص.....	xv
BAB I.....	1
1.1. Latar Belakang	1
1.2. Masalah Penelitian	4
1.3. Hipotesis	5
1.4. Tujuan Penelitian	5
1.5. Batasan Masalah	5
1.6. Manfaat Penelitian	5
BAB II	7
2.1. BPM	7
2.2. iBPM	9
2.3. BPMN.....	12
2.4. <i>Component Based Development</i>	16
2.5. The V Model CBD	18
2.6. XML Process Definition Language (XPDL)	22
BAB III.....	26
3.1 Gambaran Umum.....	26
3.2 Sumber Data	26
3.3.1 Data Primer	26

3.3.2 Data Sekunder	29
3.3 Metode.....	33
3.3.1 Analisis dan Spesifikasi Kebutuhan	33
3.3.2 Desain Sistem.....	34
3.3.3 Implementasi dan <i>Unit Testing</i>	34
3.3.4 Integrasi Sistem.....	34
3.3.5 Verifikasi dan Validasi Sistem	35
3.3.6 Dukungan dan Operasi Perawatan.....	35
3.4 Skenario Pengujian.....	35
3.4.1 The Yaqin Complexity.....	36
3.4.2 <i>Accuracy</i>	38
BAB IV	39
4.1 Hasil Pengembangan <i>Software</i>	39
4.4.1 Spesifikasi dan <i>Analisis</i> Kebutuhan.....	39
4.4.2 Desain Sistem	43
4.4.3 Implementasi dan <i>Unit Testing</i>	44
4.4.4 Integrasi Sistem.....	47
4.4.5 Verifikasi dan Validasi Sistem	51
4.2 Hasil Eksperimen	52
4.2.1 Kompleksitas <i>Workflow</i>	52
4.2.2 Akurasi <i>Web Service Discovery</i>	57
4.3 Pembahasan	60
4.4 Integrasi Sains dan Islam.....	61
BAB V.....	63
5.1 Kesimpulan	63
5.2 Saran	64
DAFTAR PUSAKA	65

DAFTAR GAMBAR

BAB II

Gambar 2. 1 Sikus hidup menurut BPM Van der Aalst et al	8
Gambar 2. 2 CBD Model : Kiri ke kanan laBagan 1lu bawah, Y Model, X Model dan CBSD Dual Life Cycle Model.....	18
Gambar 2. 3 Proses pengembangan CBD Model V	19
Gambar 2. 4 Contoh sintaks XPDL	24

BAB III

Gambar 3. 1 Prosedur penelitian	33
---------------------------------------	----

BAB IV

Gambar 4. 1 Arsitektur komponen <i>workflow repository</i>	39
Gambar 4. 2 XML dari RDF	40
Gambar 4. 3 Arsitektur komponen <i>web service discovey</i>	41
Gambar 4. 4 Arsitektur komponen sistem BPM	42
Gambar 4. 5 GUI manajemen <i>web service</i>	46
Gambar 4. 6 GUI <i>upload XPDL</i>	47
Gambar 4. 7 GUI pemetaan <i>web service</i>	47
Gambar 4. 8 UML Class <i>WorkflowLib</i>	48
Gambar 4. 9 UML Class <i>EncapLib</i>	49
Gambar 4. 10 UML class <i>TenantLib</i>	49
Gambar 4. 11 UML class <i>WSLib</i>	49
Gambar 4. 12 Database 3 komponen	50
Gambar 4. 13 GUI integrasi <i>workflow repository</i>	50
Gambar 4. 14 GUI integrasi <i>web service discovery</i>	51
Gambar 4. 15 <i>workflow</i> tambah <i>user</i>	52
Gambar 4. 16. <i>workflow</i> tambah proerti baru.....	53
Gambar 4. 17. <i>workflow</i> manajemen <i>user</i>	53

DAFTAR TABEL

BAB II

Tabel 2. 1 <i>Literatur review</i> iBPM.....	11
Tabel 2. 2 Notasi <i>flow object</i>	13
Tabel 2. 3 Notasi <i>connecting object</i>	14
Tabel 2. 4 Notasi <i>swimlanes</i>	15
Tabel 2. 5 Notasi <i>artifacts</i>	15
Tabel 2. 6 Model pada CBD.....	17

BAB IV

Tabel 4. 1 Hasil analisis <i>source code</i> komponen.....	43
Tabel 4. 2 Hasil unit <i>testing workflow repository</i>	45
Tabel 4. 3 Hasil unit <i>testing web service discovery</i>	45
Tabel 4. 4 Tabel <i>routing database</i>	48
Tabel 4. 5 Hasil validasi <i>workflow repository</i>	51
Tabel 4. 6 Hasil validasi <i>web service discover</i>	51
Tabel 4. 7 Perhitungan The Yaqin Complexity.....	54
Tabel 4. 8 Perbandingan jumlah klik	55
Tabel 4. 9 Hasil hitung <i>accuracy web service discovery</i> sebelum integrasi	57
Tabel 4. 10 Hasil hitung <i>accuracy web service discovery</i> setelah integrasi	58

ABSTRAK

Fardana, Fikri Fiuca. 2021. ***PENGEMBANGAN SISTEM PENGLOLAAN PROSES BISNIS CERDAS***. Skripsi. Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Muhammad Ainul Yaqin, M.Kom. (II) Syahiduz Zaman, M.Kom.

Kata Kunci : *Workflow Repository, Web Service Discovery, BPM, iBPM*

Penelitian ini bertujuan membuat mengetahui perbedaan waktu yang diperlukan untuk konfigurasi BPM Sistem sebelum dan sesudah diintegrasikan dengan *artificial intelligence*. Integrasi ini diharapkan bisa memberi kelebihan pada BPM Sistem yaitu otomatisasi dan manajemen workflow. Penelitian ini mengintegrasikan BPM Sistem dengan komponen *workflow repository* menggunakan *ontology* dan *web service discovery* menggunakan *cosine similarity*. Penelitian ini akan menggunakan metode *Component Based Development*, dimana dari beberapa metode tersebut, model yang akan dipakai adalah *V Model*. Hasil penelitian menunjukkan bahwa BPM Sistem sesudah integrasi memerlukan waktu lebih sedikit dibanding dengan integrasi sebelum integrasi. Hasil ini dapat diketahui dari nilai rasio perbandingan dan peningkatan yang lebih sedikit setelah integrasi.

ABSTRACT

Fardana, Fikri Fiuca. 2021. ***DEVELOPMENT OF INTELLIGENT BUSINESS PROCESS MANAGEMENT SYSTEM***. Essay. Department of Informatics, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Counselo: (I) Muhammad Ainul Yaqin, M.Kom. (II) Syahiduz Zaman, M.Kom.

Keywords : *Workflow Repository, Web Service Discovery, BPM, iBPM*

This study aims to determine the difference in time required to configure the BPM System before and after integration with artificial intelligence. This integration is expected to provide advantages to the BPM System, namely automation and workflow management. This research integrates BPM System with workflow repository components using ontology and web service discovery using cosine similarity. This research will use the Component Based Development method, where from some of these methods, the model to be used is the V Model. The results showed that the BPM System after integration requires less time than integration before integration. This result can be seen from the ratio value and the increase is less after integration.

الملخص

فاررانا، فكري فيوجا. ٢٠٢١. تطوير نظام ذكي لإدارة عمليات الأعمال. مقال. قسم المعلوماتية ، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانغ. المشرف: ١ (محمد عين اليقين ، الماجستير) ٢٠ (سياهيدوز زمان ، الماجستير).

الكلمات الرئيسية: مستودع سير العمل ، اكتشاف خدمة الويب ، BPM ، iBPM.

مستخلص البحث: تهدف هذه الدراسة إلى تحديد فرق الوقت المطلوب لتكوين نظام BPM قبل وبعد التكامل مع الذكاء الاصطناعي. من المتوقع أن يوفر هذا التكامل مزايا لنظام BPM ، وهي الأتمتة وإدارة سير العمل. يدمج هذا البحث نظام BPM مع مكونات مستودع سير العمل باستخدام علم الوجود واكتشاف خدمة الويب باستخدام تشابه جيب التمام. سيستخدم هذا البحث طريقة التطوير المبني على المكون ، حيث من بعض هذه الطرق النموذج المستخدم هو النموذج الخامس أظهرت النتائج أن نظام BPM بعد التكامل يتطلب وقتاً أقل من التكامل قبل التكامل. يمكن رؤية هذه النتيجة من قيمة النسبة وتكون الزيادة أقل بعد التكامل..

BAB I

PENDAHULUAN

1.1. Latar Belakang

Proses bisnis merupakan hal yang penting dari bisnis itu sendiri. Proses bisnis pada sebuah perusahaan tidak selalu sama, sehingga bersifat dinamis. Perubahan proses diperlukan dengan tujuan menyesuaikan dengan perkembangan lingkup bisnisnya. *Business Process Management* atau BPM diperlukan untuk mengatasi ini. BPM melibatkan aspek pemodelan, otomasi, eksekusi, pengawasan, pengukuran dan optimasi dari aktivitas bisnis sebagai pendukung tercapainya tujuan, dan melibatkan seluruh pemain aliran proses bisnis itu sendiri.

Business Process Management atau BPM adalah sistem yang komprehensif untuk mengelola dan mengubah operasi organisasi (Hammer, 2015). Pada, BPM pendekatannya adalah untuk mengelola pelaksanaan operasi bisnis yang didukung TI dari pandangan pakar bisnis dari pada bidang teknis (Hepp et al., 2005). Dengan ini, melalui manajemen proses, perusahaan dapat memastikan untuk memenuhi target serta perusahaan dapat menentukan kapan proses tidak lama dibutuhkan dan perlu diganti.

Penerapan BPM pada perusahaan memiliki keuntungan yang signifikan, karena pada BPM, pendekatannya pada proses bisnis suatu perusahaan sehingga perusahaan merespon dengan baik sesuai kondisi pada lingkup bisnisnya. Keuntungan lainnya adalah biaya operasi lebih rendah yang didapat akibat konsistensi, biaya, kecepatan, kualitas dan layanan yang lebih baik dari kemampuan sistem terhadap perusahaan.

Konsep proses bisnis yang disebut juga konsep tugas, divisi, produksi dan produk muncul di awal abad ke-20 dan berkembang sampai pada 2010 menjadi pendekatan pendekatan manajemen dan perbaikan dengan definisi formal dan pemodelan teknis. Pada tahap sekarang ini, telah berkembangnya *artificial intelligent* dapat dikombinasikan dengan BPM. Prediksi pemeliharaan dan digitalisasi alur kerja pabrik serta pelacakan objek bisnis yang cerdas dihasilkan dari kombinasi aplikasi dan teknologi modern dalam proses dengan tujuan keunggulan operasional alur kerja. Sehingga, potensi untuk model bisnis yang inovatif akan muncul (Lederer et al., 2017).

BPM telah diterapkan sampai sekarang ini. Penerapan *artificial intellegency* pada BPM akan menutupi kekurangan yang ada saat ini. Salah satu kelemahannya adalah sulit untuk pemeliharaan. Pada pemeliharaannya, jika membutuhkan perubahan bisnis proses, membutuhkan tim teknis yang handal. Hal ini juga dapat menaikkan biaya pengeluaran. Prinsip BPM adalah *less-code* dan memiliki kemampuan kostimasasi yang tinggi.

Integrasi BPM dan *artificial intellegency* memeberikan keuntungan yang lebih pada optimalisasi aktivitas bisnis proses manajemen. Penerapan *artificial intelligence* dapat mengotomasasi proses pada proses bisnis proses manajemen. Secara tidak langsung, menyediakan kemampuan pada efektifitas dan fleksibilitas manajemen bisnis (Shi & Lu, 2010). Hal ini sama halnya dengan mengurangi kebutuhan *coding* pada pemeliharaannya dan meningkatkan kemampuan kostumisasi BPM.

BPM pada dasarnya memiliki tiga bagian utama untuk menjalankan sebuah file bisnis proses. Tiga komponen itu antara lain *workflow repository*, *web service discovery*, *running environment*. Pada penelitian ini, penulis akan mengintegrasikan komponen *artificial intelligence* yaitu *workflow repository* menggunakan *ontology* (Astrininditya, 2017) dan *web service discovery* menggunakan *cosine similarity* (Dewi, 2019) ke dalam BPM. Integrasi dilakukan pada BPM karena dapat dilakukan opimalisasi dan otomasi fungsionalitas pada bagian BPM tersebut.

Islam merupakan agama yang sempurna dengan segala ketentuan-ketentuannya yang telah di atur di dalam Al-Quran dan Hadits Rasulullah SAW. Islam dibawa oleh Rasulullah SAW dan diajarkan kepada umatnya sebagai rahmatan lil' alamin. Ajaran Islam telah membawa kita berubah dari masa jahiliah ke masa yang mengenal aqidah dan akhlak. Dengan Al-Quran dan Sunnah Rasulullah SAW terdapat petunjuk untuk menjalani hidup. Sebagai seorang muslim yang taat dan beriman kepada Allah SWT, hendaklah memiliki akhlak yang sesuai dengan Al-Quran dan Sunnah, sehingga kita bisa kembali ke kehidupan akhir yaitu kehidupan akhirat dengan keadaan selamat dan bahagia. Pada Al-Quran, terdapat sebuah ayat pada Surah Al Isra' ayat 19 yang berbunyi :

وَمَنْ أَرَادَ الْآخِرَةَ وَسَعَىٰ لَهَا سَعْيَهَا وَهُوَ مُؤْمِنٌ فَأُولَٰئِكَ كَانَ سَعْيُهُمْ مَّشْكُورًا ۝١٩

“ Dan barangsiapa yang menghendaki kehidupan akhirat dan berusaha ke arah itu dengan sungguh-sungguh sedang ia adalah mukmin, maka mereka itu adalah orang-orang yang usahanya dibalasi dengan baik.” (QS. Al Isra' : 19)

Ayat tersebut memiliki tafsir dari kitab tafsir jalalayn sebagai berikut, “Dan barang siapa yang menghendaki kehidupan akhirat dan berusaha ke arah itu dengan sungguh-sungguh) yakni ia beramal dengan amal yang dengannya ia berhak untuk mendapatkan kehidupan akhirat (sedangkan ia adalah mukmin) kalimat ini berkedudukan menjadi hal (maka mereka itu adalah orang-orang yang usahanya dibalas dengan baik) di sisi Allah; artinya amalnya diterima oleh-Nya dan mendapat pahala dari-Nya” (AL-Mahali et al., 2007).

Dalam QS. Al Isra’ di atas menunjukkan bahwa seseorang yang bersungguh-sungguh pada suatu harapannya dan ke arah yang lebih baik, akan menuai hasil dari kebbaikannya. Ayat tersebut menunjukkan kepada kita agar menjadi seseorang yang selalu berusaha lebih baik. Berkaca dari ayat tersebut, kita bisa berusaha dengan memperbaiki lagi suatu hal yang ada sebelumnya. Melakukan perbaikan pada sebuah sistem salah satunya. Hasil dari perbaikan adalah kemanfaatan untuk yang pemakainya, serta harap lebih kepada orang-orang sekitar pemakainya. Tidak menutup kemungkinan pada prosesnya, mengalami kendala-kendala dan membutuhkan usaha lebih untuk menyelesaikannya. Oleh sebab itu, kewajiban kita untuk bersungguh-sungguh dan berdoa serta menyerahkan hasil kepada Alla SWT.

1.2. Masalah Penelitian

Masalah penelitian pada penulisan ini adalah sebagai berikut :

1. Bagaimana agar waktu yang diperlukan untuk konfigurasi lebih cepat pada *workflow* komplekstitas tinggi ?
2. Bagaimana meningkatkan akurasi pencarian *web service*?

1.3. Hipotesis

Hipotesis penelitian pada penulisan ini adalah sebagai berikut :

1. sistem BPM dengan menggunakan *intelligent system* akan mempercepat waktu konfigurasi

1.4. Tujuan Penelitian

Tujuan penelitian pada penulisan ini adalah sebagai berikut :

1. Mengembangkan sistem BPM yang cerdas untuk mempercepat waktu konfigurasi dengan *workflow* kompleksitas tinggi.
2. Mengembangkan sistem BPM yang cerdas untuk meningkatkan akurasi pencarian *web service*.

1.5. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Data uji digunakan untuk *testing* komponen *artificial intelligence* adalah data sampel penelitian sebelumnya yaitu penelitian dengan judul *Web Service Discovery* menggunakan *cosine similarity* untuk meningkatkan akurasi *query* pada *Web Service Repository* (Astrininditya, 2017), *Workflow Repository* pada ERP Pondok Pesantren menggunakan *ontology* untuk meningkatkan *reusability* (Dewi, 2019) dan Analisis dan Perancangan *Software* BPM Sistem (Shomad Bukhori, 2020).
2. Data uji *workflow* mengikuti batasan yang ditetapkan pada penelitian BPM System sebelumnya.

1.6. Manfaat Penelitian

Manfaat penelitian pada penelitian ini adalah sebagai berikut :

1. Meningkatkan akurasi pencarian *web service discovery*.
2. Mengkonfigurasi dengan waktu lebih cepat dengan kompleksitas *workflow* tinggi.

BAB II

STUDI LITERATUR

2.1. BPM

Manajemen proses bisnis (BPM) adalah pendekatan untuk mengelola pelaksanaan operasi bisnis yang didukung TI dari sudut pandang pakar bisnis alih-alih dari sudut pandang teknis (Hepp et al., 2005). BPM berkaitan dengan mengelola perubahan untuk meningkatkan proses bisnis. BPM menyatukan disiplin yang sebelumnya berbeda dari pemodelan proses, simulasi, alur kerja, integrasi aplikasi perusahaan (*Enterprise Application Integration* atau EAI), dan integrasi *Business to Business* (B2B) ke dalam standar tunggal (Owen & Raj, 2003).

The Business Process Management Initiative (BPMI - www.bpmi.org) telah dibentuk untuk mempromosikan dan mengembangkan penggunaan Manajemen Proses Bisnis (BPM) melalui penggunaan standar untuk desain proses, penyebaran, pelaksanaan, pemeliharaan, dan optimalisasi proses. BPMI mengembangkan tiga standar untuk memfasilitasi BPM, antara lain :

1. BPMN atau *Business Process Model and Notation*, sebagai standar untuk memodelkan proses bisnis
2. *Business Process Model Language* atau BPML, sebagai Bahasa eksekusi bisnis standar
3. *Business Process Query Language* atau BPEL, sebagai antarmuka manajemen standar untuk penyebaran dan pelaksanaan e-bisnis proses.

Pemahaman terhadap terminologi dan fitur BPM, dapat dimulai dari siklus BPM itu sendiri. Terdapat banyak pandangan mengenai siklus hidup BPM,

dalam hal ini akan mengambil dari Van der Aalst *et al*, karena ini cocok dan relevan. Siklus hidup BPM menurutnya adalah terdiri dari :

1. Desain proses

Dalam tahap ini, proses bisnis apa pun berbasis faks atau kertas dimodelkan secara elektronik ke dalam sistem BPM (BPMS).

2. Konfigurasi sistem

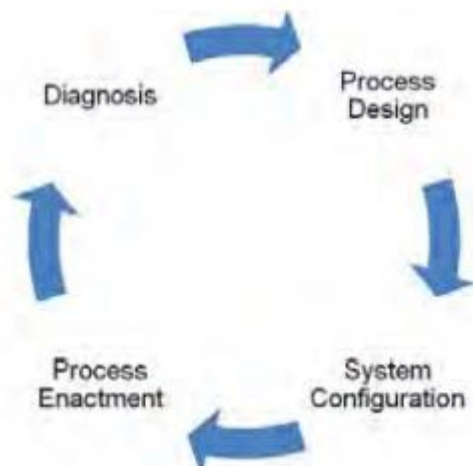
Tahap ini mengonfigurasi BPMS dan infrastruktur sistem yang mendasarinya, misalnya sinkronisasi peran dan bagan organisasi.

3. Pengesahan proses

Proses bisnis yang dimodelkan secara elektronik digunakan dalam BPMS.

4. Dagnosis

Diberikan alat analisis dan pemantauan yang tepat, analis BPM dapat mengidentifikasi dan meningkatkan kemacetan dan potensi celah penipuan dalam proses bisnis.



Gambar 2. 1 Siklus hidup menurut BPM Van der Aalst et al

2.2. iBPM

Pada decade 2010-an ini, intelligent BPM (iBPM) beberapa kali dibicarakan dan telah memberi dorongan dengan integrasi teknologi analisis ke dalam orkestra proses. Hal ini memungkinkan membuat perusahaan bisnis mereka lebih cerdas dan memberikan perusahaan mengamati situasi proses secara langsung dan kemampuan untuk merespon lebih tepat (Gao, 2013). Keunggulan seperti ini disebut juga sebagai evolusi BPM, karena beberapa alasan berikut. Pertama, akan memenuhi kebutuhan kelincahan proses yang sedang berlangsung, terutama untuk perubahan peraturan dan penanganan pengecualian yang lebih dinamis. Kedua, ini akan bertujuan meningkatkan ketersediaan data yang lebih besar dari dalam dan luar perusahaan sebagai *input* ke dalam pengambilan keputusan. Ketiga, ini akan memfasilitasi interaksi dan kolaborasi dalam proses lintas batas.

iBPM pada sudut pandang infrastruktur aplikasi dan *framework*, iBPM Suite mewarisi semua fitur BPM Suite tradisional, dilengkapi dengan teknologi yang lebih canggih. Dari sudut pandang aplikasi perusahaan dan kerja sama, perbedaan iBPM generasi berikutnya dari yang sekarang dapat diringkas sebagai “4A” berikut:

1. Analisis

Fitur iBPM yang paling menonjol adalah kemampuan analitik tingkat lanjut. Ini terintegrasi dengan teknologi analitik canggih, termasuk pra-analitik dan pasca-analitik. Yang pertama terutama berkonsentrasi pada analisis berbasis model proses, seperti model dekomposisi, deteksi klon, pencarian kesamaan dan lain-lain. Yang terakhir menggunakan *log*

historis dan informasi lainnya, dan mengacu pada penemuan proses bisnis otomatis, analisis sosial, rekomendasi cerdas, prediksi dan sebagainya.

2. Otomasisasi

Volume data yang sangat besar memerlukan teknik analisis otomatis atau semi-otomatis untuk mendeteksi pola, mengidentifikasi anomali, dan mengekstraksi pengetahuan. Contoh konsolidasi proses bisnis. Itu selalu merupakan tugas yang sangat sulit untuk organisasi besar dengan lebih dari ribuan model proses. IBPM harus dirancang untuk memfasilitasi prosedur yang secara otomatis mengurangi duplikasi dan membuat perbedaan antara model proses eksplisit, bukan operasi manual.

3. Adaptasi

Perubahan dinamis dari proses bisnis dan data eksternal di dalam dan luar harus secara fleksibel ditangkap dan direspon dengan menggunakan tidak hanya penyesuaian adaptif dari parameter algoritma analisis, tetapi juga pemilihan algoritma yang sesuai sesuai permintaan dengan cara konfigurasi

4. Kecerdasan

Terdapat celah besar, analis bisnis memiliki pemahaman yang mendalam tentang bisnis tetapi tidak dapat merancang model proses secara mandiri tanpa dukungan staf TI, meskipun bahasa pemodelan berbasis notasi dieksploitasi. IBPM diharapkan menyederhanakan prosedur. Misalnya, dengan memasukkan fragmen proses dengan

semantik bisnis ke dalam alat desain, efisiensi pemodelan dapat ditingkatkan secara signifikan dan sebagian besar prosedur dapat diimplementasikan oleh analis bisnis dengan upaya TI paling sedikit.

Tabel berikut ini merupakan beberapa penelitian terkait iBPM yang telah dilakukan sebelumnya.

Tabel 2. 1 *Literatur review iBPM*

No	Penelitian Terkait	Hasil Penelitian	Kontribusi Penelitian
1	<i>Business Process Management and Digital Innovations : A Systematic Literatur Review</i> (Ahmad & Van Looy, 2020)	DI (<i>Digital Innovations</i>) membantu menyelesaikan tugas lebih cepat dan cara lebih baik. Dengan menyajikan beberapa penelitian tentang BPM, penelitian ini telah membuka jalan untuk penelitian BPM selanjutnya. Kedepannya, praktisi akan memikirkan pemakaian teknologi transformasi keputusan tentang BPM.	Penelitian ini mengumpulkan 231 artikel akademik tentang integrasi BPM dengan DI (<i>Digital Innovations</i>). Penelitian ini bertujuan menggali keunggulan dari teknologi informasi baru, yang berguna untuk penelitian BPM dan menemukan tren sekarang pada praktiknya.
2	<i>On the Synergies Between Business Process Managemeng and Digital Innovations</i> (Van Looy, 2018)	Penelitian ini memperkuat pembahasan tentang berubahnya peran BPM pada dunia digital. Terlebihnya, disiplin BPM akan menguntungkan dari kolaborasinya dengan disiplin lain seperti <i>case-driven BPM</i> , <i>collaboration BPM</i> atau <i>intelligent BPM</i> .	Penelitian ini bertujuan menyikap berubahnya cara kerja bisnis karena meningkatnya peluang dan munculnya teknologi baru dengan cepat. Penelitian ini berfokus pada penyampaian peran BPM terhadap DI (<i>Digital Innovations</i>).
3	<i>Introduction to the 1st International Workshop on Business Process Innovation with Artificial</i>	Workshop mengidentifikasi banyak sumber untuk kolaborasinya antara AI dan BPM. Dalam beberapa kasus, beberapa solusi AI dapat diimplementasikan pada BPM. Terlebihnya, diskusi berakhir dengan	Workshop ini bertujuan untuk menyampaikan kepada pembaca bahwa AI dan BPM dapat dikolaborasikan satu sama lain. Workshop ini membuka jalan baru

	<i>Intelligence</i> (BPAI 2017) (Teniente & Weidlich, 2018)	setujunya untuk melanjutkan penelitian tentang BPM dan AI dengan goalnya adalah paper atau area penelitian lain tetapi masih dalam topik tersebut.	untuk penelitian topik tersebut kedepannya.
--	---	--	---

2.3. BPMN

Business Process Model and Notation (BPMN) adalah standar baru untuk memodelkan aliran proses bisnis dan layanan web (Owen & Raj, 2003). Dibuat oleh *Business Process Management Initiative* (BPMI), tujuan awal BPMN adalah untuk memberikan notasi yang dapat dipahami oleh semua pelaku bisnis. Ini termasuk analisis bisnis yang membuat konsep awal proses kepada pengembang teknis yang bertanggung jawab untuk menerapkan teknologi yang akan melakukan proses tersebut. Tujuan kedua, yang sama pentingnya adalah untuk memastikan bahwa bahasa XML yang dirancang untuk pelaksanaan proses bisnis, seperti BPEL4WS (*Business Process Execution Language for Web Service*) dan BPML (*Business Process Model Language*), dapat diekspresikan secara visual dengan notasi umum. BPMN membuat standarisasi untuk jarak antara desain proses bisnis dan proses implementasi.

Pemodelan proses bisnis digunakan untuk mengkomunikasikan berbagai informasi ke pengguna yang berbeda. BPMN dirancang untuk mencakup banyak jenis pemodelan dan memungkinkan penciptaan segmen proses serta proses bisnis ujung-ke-ujung, pada berbagai tingkat struktural. Di dalam berbagai tujuan pemodelan proses, ada dua tipe dasar model yang dapat dibuat dengan BPD (*Business Process Diagram*) :

1. *Collaborative (Public) B2B Processes*

BPD ini menggambarkan interaksi antara dua atau lebih entitas bisnis. Diagram untuk jenis proses ini umumnya dari sudut pandang global. Mereka tidak mengambil pandangan dari peserta tertentu, tetapi menunjukkan interaksi antara para peserta. Interaksi digambarkan sebagai urutan kegiatan dan pola pertukaran pesan antara peserta.

2. *Internal (Private) Business Processes*


Proses bisnis ini akan fokus pada sudut pandang organisasi bisnis tunggal. BPD yang digambarkan juga memungkinkan entitas luar, tetapi ini akan didefinisikan sebagai proses umum, bukan proses secara rinci.



BPD dibuat dari kumpulan beberapa elemen notasi. Terdapat banyak notasi pada BPMN, seperti aktivitas yang digambarkan sebagai kotak, dan percabangan yang digambarkan sebagai jajar genjang. Elemen-elemen tersebut dapat dikelompokkan ke empat kategori dasar :

1. Objek aliran atau *flow object*

BPD memiliki tiga elemen inti dari beberapa elemen yang ada pada BPMN untuk menggambarkan aliran objeknya. Berikut tabel penjelasannya :

Tabel 2. 2 Notasi *flow object*



Elemen	Deskripsi	Notasi
<i>Event</i>	Peristiwa ini mempengaruhi aliran proses dan biasanya memiliki sebab (pemicu) atau akibat (hasil). <i>Event</i> adalah lingkaran dengan pusat terbuka untuk memungkinkan penanda internal untuk membedakan pemicu atau hasil yang berbeda.	


<i>Activity</i>	Suatu kegiatan diwakili oleh persegi panjang sudut bundar dan merupakan istilah umum untuk pekerjaan yang dilakukan perusahaan. Jenis-jenis dari <i>activity</i> adalah <i>Task</i> dan <i>Sub-Proses</i> . <i>Sub-Proses</i> dibedakan dengan tanda tambah kecil di tengah bawah bentuk	
<i>Gateway</i>	<i>Gateway</i> digunakan untuk menentukan keputusan, serta percabangan, penggabungan, dan penyatuan jalan. <i>Marker</i> penanda akan menunjukkan jenis kontrol perilaku pada <i>gateway</i> .	

2. Objek Penghubung atau *connecting object*

Object flow dihubungkan bersama dalam diagram untuk membuat struktur kerangka dasar dari proses bisnis. Ada tiga Objek Penghubung yang menyediakan fungsi ini. Penghubung ini disebutkan sebagai berikut :

Tabel 2. 3 Notasi *connecting object*



Elemen	Deskripsi	Notasi
<i>Sequence Flow</i>	<i>Sequance flow</i> digunakan untuk menunjukkan kemana proses akan dieksekusi selanjutnya.	
<i>Message Flow</i>	Menunjukkan aliran pesan antara dua partisipan, siapa yang mengirim dan menerima pesan.	
<i>Association</i>	Digunakan untuk menunjukan kumpulan data, <i>text</i> dan metadata kemana mengalir. <i>Association</i> menggambarkan	

	<i>input</i> dan <i>output</i> suatu aktivitas.	
--	---	---

3. *Swimlanes*

Swimlanes digunakan untuk memisahkan dan membedakan *activity* sesuai hak akses pada suatu perusahaan atau pada kasus B2B menggambarkan perusahaan-perusahaan yang terlibat. Contohnya pada perusahaan bisa dianalogikan sebagai *level* jabatan yang berbeda. Pada BPMN terdapat dua notasi swimlanes, seperti tabel berikut :



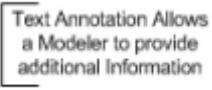
Tabel 2. 4 Notasi *swimlanes*

Elemen	Deskripsi	Notasi
<i>Pool</i>	Menggambarkan sebuah partisipan pada proses. <i>Pool</i> bisa dianalogikan seperti satu perusahaan. <i>Multipool</i> biasanya menggambarkan konteks B2B diagram.	
<i>Lane</i>	<i>Lane</i> adalah sub-divisi pada sebuah <i>pool</i> . <i>Lane</i> digunakan untuk mengorganisasi dan mengkatogorikan aktivitas.	

4. *Artifacts*

Beberapa kasus, BPMN diharuskan fleksibilitas pada notasi dasarnya untuk memodelkan aliran data. Sejumlah artifacts dapat ditambahkan ke diagram, yang sesuai untuk konteks proses bisnis tertentu. Terdapat tiga artifak pada BPM, yang dijelaskan sebagai berikut :

Tabel 2. 5 Notasi *artifacts*

Elemen	Deskripsi	Notasi
<i>Data Object</i>	Adalah mekanisme untuk menggambarkan bagaimana data dibutuhkan atau dibuat oleh aktivitas. <i>Data object</i> dan aktivitas dihubungkan dengan <i>association</i> .	
<i>Group</i>	Digunakan untuk tujuan dokumentasi atau analisi dan tidak mempengaruhi aliran proses.	
<i>Annotation</i>	Diguakan untuk menyediakan informasi tambahan untuk penggunaannya.	

2.4. Component Based Development

Component-based software engineering (CBSE), atau juga disebut *components-based development* (CBD) adalah cabang rekayasa perangkat lunak yang menekankan independensi dari setiap satuan sehubungan dengan fungsionalitasnya terhadap sistem perangkat lunak (“Component-Based Software Engineering,” n.d.). CBSE berkaitan dengan seluruh siklus hidup produk berbasis komponen telah difokuskan pada teknologi yang terkait dengan desain dan implementasi komponen perangkat lunak dan sistem yang dibangun dari komponen perangkat lunak (Crnkovic et al., 2005).

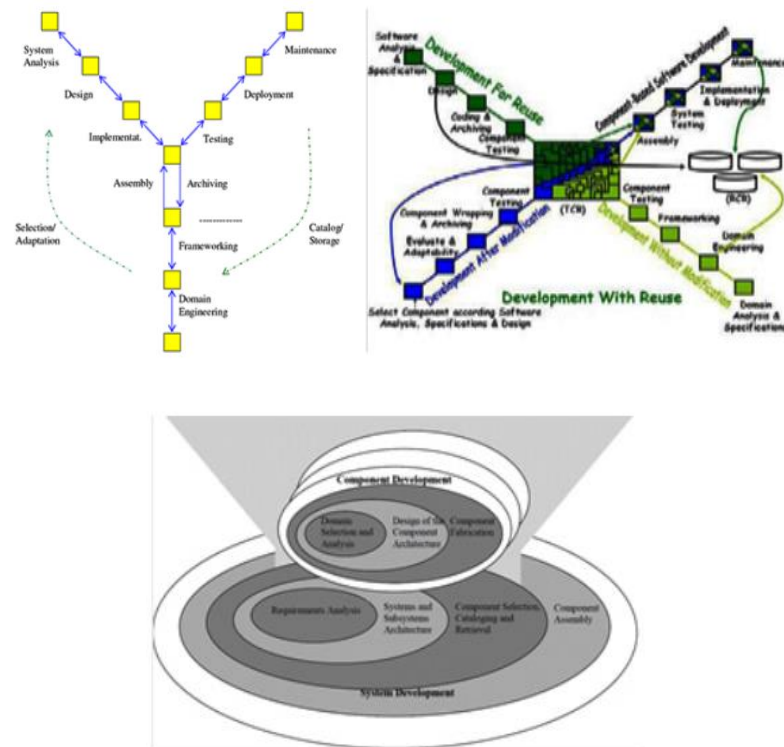
Gagasan utama dari CBD adalah membangun sistem dari komponen yang sudah ada sebelumnya. Asumsi ini memiliki beberapa konsekuensi untuk siklus hidup sistem. Pertama, proses pengembangan sistem berbasis komponen dipisahkan dari proses pengembangan komponen; komponen harus sudah

dikembangkan dan mungkin digunakan dalam produk lain ketika proses pengembangan sistem dimulai. Kedua, proses baru yang terpisah akan muncul. Menemukan dan mengevaluasi komponen. Ketiga, kegiatan dalam proses akan berbeda dari kegiatan dalam pendekatan non-komponen; untuk pengembangan sistem penekanannya akan pada menemukan komponen yang tepat dan memverifikasi mereka, dan untuk pengembangan komponen, desain untuk digunakan kembali akan menjadi perhatian utama.

CBD sendiri mempunyai beberapa model yang diperkenalkan beberapa tahun belakangan ini. Terdapat enam model untuk sekarang, diawali dari Y model diperkenalkan tahun 2005 sampai yang terbaru, W Model yang diperkenalkan tahun 2011 (Gulia & Palak, 2017). Tabel berikut akan menjelaskan secara singkat setiap model dari CBD :

Tabel 2. 6 Model pada CBD

CBD Model	Tahun	Fokus Area
Y Model	2005	Reusability dan pengembangan parallel
V Model	2005	Pengembangan komponen dan pemilihan komponen
X Model	2008	Pengembangan komponen dan pengembangan system dengan atau tanpa modifikasi komponen
CBSD Dual Life Cycle Model	2009	Pemisahan antara pengembangan komponen dan pengembangan sistem
Knot Model	2011	Reusability, modularitas, dan resiko analisis
W Model	2011	Verifikasi dan validasi, pemisahan komponen dan pengembangan sistem

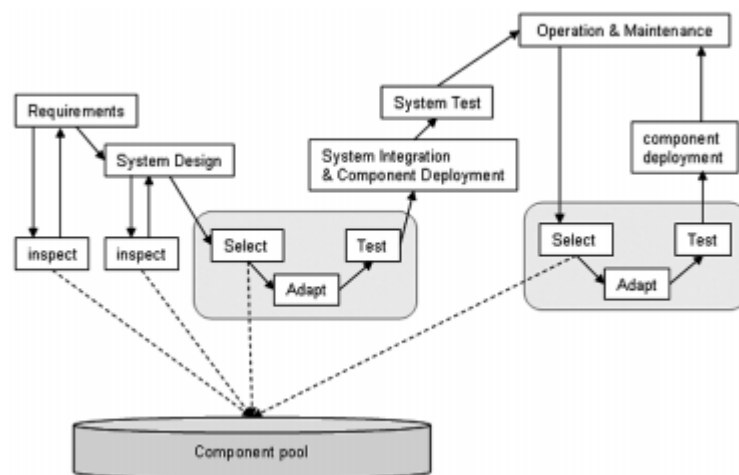


Gambar 2. 2 CBD Model : Kiri ke kanan laBagan 1lu bawah, Y Model, X Model
dan CBSD Dual Life Cycle Model

2.5. The V Model CBD

Model V mengadopsi pendekatan pengembangan perangkat lunak tradisional untuk membangun sistem dari komponen perangkat lunak yang dapat digunakan kembali. Kegiatan pengembangan ini terdiri dari beberapa langkah dan memberikan informasi detail pada fase desain. Penekanan utama V-Development adalah siklus pengembangan komponen (Khan & Khan, 2012). Daur hidup pengembangan komponen dianggap sebagai proses yang berbeda. Tahap seleksi mendapat *input* dari sistem terpisah yang biasanya menemukan dan mengevaluasi komponen yang sesuai untuk dikomposisikan ke dalam sistem. Model V adalah adaptasi dari model air terjun tradisional yang kaku untuk pengembangan sistem modular dengan sedikit fleksibilitas. Setiap

fase harus diselesaikan sebelum fase berikutnya dimulai. Pengujian ditekankan dalam model ini lebih dari model air terjun. Prosedur pengujian dikembangkan pada awal siklus hidup sebelum pengkodean dilakukan, selama masing-masing fase sebelum implementasi. Persyaratan memulai model siklus hidup seperti halnya model air terjun. Sebelum pengembangan dimulai, rencana pengujian sistem dibuat. Rencana pengujian berfokus pada pemenuhan fungsi yang ditentukan dalam pengumpulan persyaratan.



Gambar 2. 3 Proses pengembangan CBD Model V

Pada gambar 2.3, terlihat bahwa proses dari pengembangan model V terdiri dari enam proses. Tahap paling awal, adalah pada tahap requirement and design, pihak dari kebutuhan sistem dan arsitek system harus memikirkan tentang ketersediaan pada kesiapan komponen yang telah ada. Penjelasan dari proses-proses model V akan dijelaskan sebagai berikut :

1. *Requirement analysis dan specification*, dalam fase ini satu kegiatan penting adalah menganalisis kemungkinan mewujudkan solusi yang akan memenuhi persyaratan ini. Dalam pendekatan berbasis komponen

ini menyiratkan bahwa perlu untuk menganalisis apakah persyaratan ini dapat dipenuhi dengan komponen yang tersedia. Karena terdapat kemungkinan bahwa tidak selalu komponen yang sesuai selalu dapat ditemukan, ada risiko bahwa komponen baru harus diimplementasikan. Untuk tetap dapat menggunakan keunggulan dari sebuah komponen itu, adalah menegosiasikan persyaratan dan memodifikasinya untuk dapat menggunakan komponen yang ada.

2. *System and software design*. Seperti pada tahap *requirement analysis*, spesifikasi dan desain sistem sangat terkait dengan ketersediaan komponen. Komponen potensial mematuhi model komponen tertentu. Orang dapat berasumsi bahwa akan mungkin untuk menggunakan komponen yang diimplementasikan dalam teknologi komponen yang berbeda, tetapi dalam praktiknya sangat sulit untuk mencapai interoperabilitas antara model komponen yang berbeda. Model komponen tertentu memerlukan kerangka kerja arsitektur tertentu, dan aplikasi seharusnya menggunakan kerangka kerja ini. Ini memiliki dampak langsung pada keputusan arsitektur.
3. *Implementation and unit testing*. Pada pembangunan berbasis komponen, sebenarnya adalah menghubungkan komponen secara langsung dan terintegrasi. Dalam praktiknya pengintegrasian juga akan mencakup adaptasi komponen, dan bahkan penerapan fungsi baru. Dalam kasus umum, komponen itu sendiri sudah dibangun dan diuji. Namun, tes komponen secara terpisah tidak cukup. Seringkali, unit desain akan diimplementasikan sebagai rakitan beberapa komponen.

Rakitan ini harus diuji secara terpisah karena rakitan komponen yang benar mungkin salah, meskipun komponen itu sendiri sudah benar

4. *System Integration*. Proses integrasi meliputi integrasi komponen infrastruktur standar yang membangun kerangka kerja komponen dan komponen aplikasi. Integrasi komponen tertentu ke dalam sistem disebut penyebaran komponen. Berbeda dengan keseluruhan integrasi sistem, penyebaran komponen adalah mekanisme untuk integrasi komponen tertentu pendaftaran komponen.
5. *System verification and validation*. Masalah khusus untuk pendekatan berbasis komponen adalah pada lokasi kesalahan, terutama ketika komponen dari jenis *black box software testing* dan dikirim dari vendor yang berbeda. Biasanya, komponen dapat menunjukkan kesalahan, tetapi penyebab kerusakan terletak pada komponen lain. Standarisasi antarmuka dari pihak peminata memainkan peran penting dalam memeriksa *input* dan *output* yang tepat dari komponen. Antarmuka ini memungkinkan spesifikasi *input* dan *output* dan memeriksa kebenaran data.
6. *Operation support and maintenance*. Proses pemeliharaan memiliki beberapa step yang serupa pada proses integrasi. Sebuah komponen baru atau modifikasi dari komponen akan dimasukkan ke system. Ini juga memungkinkan mengubah cara pengintegrasian antara komponen bersangkutan dengan komponen lainnya

Pada perbandingan dengan pendekatan berdasarkan bukan komponen, pada proses dengan pendekatan berdasarkan komponen lebih tidak membutuhkan

upaya lebih pada pemrograman, tetapi verifikasi dan *testing* membutuhkan pertimbangan usaha lebih. Proses verifikasi mengulang pada beberapa prosesnya dengan sedikit perbedaan pada targetnya. Pengulangan terdapat pada verifikasi komponen itu sendiri, verifikasi komponen pada pengintegrasian dan verifikasi komponen ketika telah diimplementasikan ke sistem.

2.6. XML Process Definition Language (XPDL)

Workflow telah berkembang selama beberapa dekade dan seiring berkembangnya teknologi juga telah berubah secara cepat. *XML Process Definition Language* adalah Bahasa yang diusulkan oleh *Workflow Management Coalition* (WfMC) untuk mempertukarkan definisi proses antar berbagai produk alur kerja (Wil M.P. van der Aalst, 2003). Tujuan XPDL adalah untuk menyediakan *lingua franca* atau bahasa penghubung untuk domain alur kerja yang memungkinkan definisi proses impor dan ekspor antara berbagai alat mulai dari system manajemen alur kerja hingga alat pemodelan dan simulasi.

XPDL membuat simple kumpulan dokumen dari manajemen dan membuat pemrosesan dokumen lebih handal. Dengan membuat deskripsi untuk kelas dokumen, daripada mengandalkan dokumen untuk menautkan ke pemrosesan deskripsi sendiri, XPDL memungkinkan untuk bergerak melampaui model monoliton hari ini dan untuk menambah sumber daya baru, seperti skema, style sheet, dan memproses informasi untuk konsep dari kelas dokumen. Tujuan XPDL akan dijabarkan sebagai berikut (Xiao Ying et al., 2004):

1. Menyederhanakan manajemen dokumen dengan memindahkan informasi pemrosesan dokumen dari dokumen itu sendiri ke sumber. Menghilangkan masalah keandalan yang disebabkan oleh XML 1.0 menggunakan subset internal dan opsi untuk parser yang tidak valid dengan menyediakan alat untuk menentukan pemrosesan eksternal yang lebih mudah dikendalikan.
2. Mendukung penggunaan berbagai skema kosakata, mulai dari XML 1.0 DTD (*Document Type Definition* – menggambarkan struktur, elemen dan atribut XML) hingga RDF (*Resource Description Framework* – standar pertukaran data pada web) hingga alat yang masih dalam pengembangan.
3. Mengizinkan pengembang membangun pengolahan yang lebih efisien dengan memasukkan fitur (seperti pemrosesan entitas) hanya jika diperlukan.
4. Menyediakan ruang untuk ekstensi, sehingga informasi pemrosesan khusus aplikasi dapat dimasukkan tanpa mengganggu aplikasi lain yang memproses dokumen.

```

1 <WorkflowProcess Id="Parallel">
2   <ProcessHeader DurationUnit="Y"/>
3   <Activities>
4     <Activity Id="A">
5       ...
6       <TransitionRestrictions>
7         <TransitionRestriction>
8           <Split Type="AND">
9             <TransitionRefs>
10              <TransitionRef Id="B"/>
11              <TransitionRef Id="C"/>
12            </TransitionRefs>
13          </Split>
14        </TransitionRestriction>
15      </TransitionRestrictions>
16    </Activity>
17    <Activity Id="B">
18      ...
19    </Activity>
20    <Activity Id="C">
21      ...
22    </Activity>
23    <Activity Id="D">
24      ...
25      <TransitionRestrictions>
26        <TransitionRestriction>
27          <Join Type="AND"/>
28        </TransitionRestriction>
29      </TransitionRestrictions>
30    </Activity>
31  </Activities>
32  <Transitions>
33    <Transition Id="AB" From="A" To="B"/>
34    <Transition Id="AC" From="A" To="C"/>
35    <Transition Id="BD" From="B" To="D"/>
36    <Transition Id="CD" From="C" To="D"/>
37  </Transitions>
38 </WorkflowProcess>

```

Gambar 2. 4 Contoh sintaks XPDL

XPDL dirancang untuk bertukar definisi proses, baik grafis dan semantik dari proses bisnis sebuah *workflow*. XPDL saat ini adalah format terbaik untuk pertukaran diagram BPMN dan dirancang khusus untuk menyimpan semua aspek dari BPMN. Atau dalam kata lain XPDL adalah serialisasi XML dari BPMN. XPDL mengandung unsur untuk menahan informasi grafis seperti posisi node X dan Y, serta aspek eksekusi yang akan digunakan untuk menjalankan proses. Menurut Handayani et al elemen dari XPDL ada enam, yakni:

1. Elemen <Package> adalah paket yang mendefinisikan proses bisnis. Konsep paket ini berguna untuk mengurangi definisi redundancy yang dapat digunakan untuk berbagai macam proses.
2. Elemen <WorkflowProcess> yaitu elemen yang digunakan untuk mendefinisikan sub proses dari suatu proses tertentu.

3. Elemen <ActivitySet> menunjukkan kumpulan dari berbagai macam aktivitas dan transisi yang menghubungkan antar aktivitas tersebut.
4. Elemen <BlockActivity> berfungsi untuk melakukan eksekusi atas elemen <ActivitySet>
5. Elemen <Participant> digunakan untuk menspesifikasikan partisipan pada *workflow*, misalkan entitas yang dapat mengeksekusi aktivitas.
6. Elemen <DataType> dan <DataField> digunakan untuk menspesifikasikan data relevan *workflow*. Data ini digunakan untuk penentuan keputusan atau mereferensikan data di luar dari *workflow* dan ditransfer ke semua aktivitas dan *subflows*.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Gambaran Umum

Pada penelitian ini, sistem yang dibangun adalah pembangunan *business process manajemen system* beserta beberapa komponen *artificial intelligence* yang terintegrasi pada sistemnya. Penelitian akan menggunakan metode *components based development* pada pembangunannya. *Business process manajemen system* pada penelitian ini telah dibangun pada penelitian sebelumnya tanpa integrasi *artificial intelligence* di dalamnya. Beberapa komponen *artificial intelligence* yang akan diintegrasikan juga telah dibangun pada penelitian sebelumnya. Beberapa komponen kecerdasannya adalah *web service discovery* dan *workflow repository*. *Web service discovery* menggunakan *cosine similarity* untuk meningkatkan *query* pada *web service discovery*. *Workflow repository* menggunakan *ontology* untuk meningkatkan performa *reusability*-nya. Hasil dari penelitian ini adalah sistem BPM dengan kecerdasan buatan yang keefektifannya akan dibandingkan dengan sistem BPM tanpa menggunakan kecerdasan buatan.

3.2 Sumber Data

3.3.1 Data Primer

Data primer adalah data yang akan digunakan untuk skenario pengujian. Data primer pada penelitian akan dibuat sendiri oleh peneliti. Data primer dibuat dengan kriteria sendiri tergantung pada komponen kecerdasan yang nanti akan diuji. Data uji penelitian komponen *workflow repository* akan dibuat

dengan struktur *workflow* tipe *sequence* dan *workflow* tipe *branch*. Data uji penelitian *web service discovery* akan dibuat dengan kata kunci baku dan tidak baku. Data uji *workflow repository* berjumlah sebanyak 11 data seperti rincian terdapat pada Tabel 3.1. Data uji *web service discovery* berjumlah sebanyak 51 data seperti rincian terdapat pada Tabel 3.2.

Tabel 3. 1 Data uji *workflow repository*

No.	Workflow	Activity
1	Milih	Proses Bagi Proses Kurang Tampilkan Hasil Proses Tambah <i>Input Y</i> Proses Kali <i>Input X</i>
2	Tambah User	Tambah Pengguna Akses
3	Penambahan Properti	Tambah Properti Disetujui penambahan Tambah data kamar Aktifkan property
4	Manajemen User	Masuk akses <i>level</i> Tambah akses pengguna ubah data akses pengguna Tampil daftar pengguna Hapus akses pengguna
5	Check Pengeluaran	Pilih property dan bulan Check data pengeluaran
6	Check Uang Cash	Pilih bulan Pilih Properti tandai uang bayar langsung
7	Tambah pengeluaran	Pilih property pilih bulan masukkan data pengeluaran
8	Laporan	Cek bayar pemesanan edit data pemesanan penetapan bagi hasil transfer ke mitra kirim bukti penerimaan
9	Ganti harga	Pilih property dan bulan pilih tanggal tetapkan harga baru
10	Pembatalan pemesanan	Pengajuan pembatalan persetujuan pembatalan
11	Tambah booking	Pilih properti booking isi data kamar pemesanan isi data tamu isi data pesan bayar verifikasi pembayaran

Tabel 3. 2 Data uji *web service discovery*

No	Nama	Kata kunci
1	<i>Input Variabel X</i>	<i>input_x</i>
2	<i>Input Variabel Y</i>	<i>input_y</i>
3	Proses Penambahan	proses tambah sum aritmatika
4	Menampilkan Hasil Penambahan 2 Variabel	tampil hasil aritmatika tambah 2 variabel

5	Proses Pengurangan	proses kurang min aritmatika
6	Proses Pembagian	proses bagi aritmatika
7	Proses Perkalian	proses kali aritmatika
8	View List User	tampil daftar pengguna manajemen akses
9	Add User	tambah pengguna manajemen akses
10	Delete User	hapus pengguna manajemen akses
11	Edit User	ubah pengguna manajemen akses
12	View List Property	tampil properti daftar
13	Add Property	tambah simpan properti
14	Delete Property	hapus properti
15	Edit Property	ubah properti
16	View Finance Report	tampil laporan keuangan
17	View List Booking	tampil daftar pemesanan kamar
18	Cancel Booking	batal pemesanan kamar
19	Reschedule Booking	ubah tanggal pemesanan kamar
20	Add Booking	tambah simpan pemesanan kamar
21	Select Channel	pilih cluster kategori jalur pemesanan
22	Fill Data Guest	isi data tamu pemesanan
23	Fill Data Booking	isi data kamar pemesanan
24	Save Booking	simpan data pemesanan kamar
25	View List Payment	tampil daftar pembayaran
26	Verify Payment	verifikasi pemesanan kamar pembayaran
27	Skip Proses	lewati proses
28	Setelah DIbagi Kali 10	bagi dan atau lalu kali sepuluh 10
29	Web Service Booking <i>Testing</i>	pengujian pemesanan
30	persetujuan tambah properti	setuju tambah properti
31	tampil hasil variabel	tampil hasil variabel
32	Form <i>Input</i> angka 1	isi angka pertama data
33	kalikan angka 5	kali angka lima
34	kali angka dengan <i>input</i>	kali angka diisi
35	bagi angka 2	bagi angka dua
36	tambah angka 2	tambah angka dua
37	kurang angka isi	kurang angka <i>input</i>
38	kurang angka 2	kurang angka dua
39	isi data pemesanan	isi data pesan
40	pilih metode pembayaran	pilih metode bayar
41	bayar satu	bayar metode satu
42	bayar 2	bayar metode dua
43	bayar 3	bayar metode tiga
44	verifikasi pembayaran	verifikasi bayar berhasil
45	pilih properting pemesanan	pilih properti pesan
46	pembayaran	bayar
47	active property	aktif properti buka
48	deactive property	tutup properti mati
49	tambah data kamar	tambah data kamar

50	masuk akses	masuk <i>level</i> akses
51	isi detail kamar booking	isi data kamar pesan

3.3.2 Data Sekunder

Data sekunder merupakan data semua component yang akan diintegrasikan. Data sekunder terdiri dari data komponen sistem BPM, komponen *workflow repository* menggunakan *ontology* (Astrininditya, 2017), dan komponen *web service discovery* menggunakan *cosine similarity* (Dewi, 2019). Data setiap komponen tersebut berupa *source code* dan laporan percobaan.

Berikut rincian data sekunder pada penelitian :

1. Komponen *Web Service Discovery*

Source code penelitian dengan judul *Web Service Discovery* menggunakan *cosine similarity* untuk meningkatkan akurasi *query* pada *Web Service Repository*. Penelitian ini dikerjakan oleh saudari Ratih Mayllia Dewi dua tahun sebelumnya. Hasil dari penelitian ini adalah pengujian dan analisis yang telah dilakukan terhadap 20 *query* pengujian dengan membandingkan antara SQL *query* dan *cosine similarity*, diperoleh peningkatan akurasi *query* sebesar sebesar 7.01% dari 90.18% menjadi 97.20%. Tingkat akurasi *query* yang dihasilkan lebih relevan menggunakan *cosine similarity* daripada menggunakan SQL *query*, karena komputasi dari nilai *similarity web service*, fungsi cosine memiliki kinerja lebih efisien daripada SQL *query* (Dewi, 2019).

Penelitian ini menggunakan Teori *Vector Space Matching* (VCM) yang digunakan untuk menghitung kemiripan dengan mendefinisikan sebuah *vector* yang mempresenatasikan tiap dokumen *web service*, dan

sebuah *vector* yang mempresentasikan *query* menggunakan *cosine similarity*. Tahapan pertama adalah *parsing* data WSDL, lalu dilanjutkan dengan *preprocessing* WSDL yang meliputi *case folding*, *tokenizing*, *stopword removal* (pemecahan setiap kata pada kalimat), dan *stemming*. Selanjutnya dilakukan pembobotan terhadap dokumen WSDL menggunakan TF/IDF (*Term Frequency/Inverse Document Frequency*). TF/IDF adalah pembobotan yang didasarkan pada skema *bag-of-word*, di mana sebuah dokumen dapat diwakili oleh kumpulan kata-kata yang digunakan dalam dokumen tersebut. TF – IDF juga mengasumsikan bahwa jika sebuah kata penting untuk sebuah dokumen, kata tersebut harus muncul berulang kali di dokumen tersebut sedangkan kata tersebut jarang muncul di dokumen lain (Kim et al., 2019). Tahap selanjutnya dilakukan perhitungan kemiripan antara *query* dengan dokumen dengan membandingkan *vector* antar kedaunya dengan *cosine coefficient*. Setelah didapat angka *cosine coefficient*, dilakukan perhitungan *cosine similarity* dan diurutkan bobotnya dengan bobot terbesar menjadi prioritas kecocokan antara *query* dengan dokumen WSDL.

2. Komponen *Workflow Repository*

Source code penelitian dengan judul *Workflow Repository* pada ERP Pondok Pesantren menggunakan *ontology* untuk meningkatkan *reusability* (Astrininditya, 2017). Hasil dari penelitian ini adalah *Resource Definition Framework* atau RDF dapat digunakan untuk membangun *metadata*. RDF dapat digunakan jika penulisannya benar

sesuai dan dapat divalidasi oleh RDF Validator. Hasil perhitungan uji coba, didapatkan nilai *recall*, *precision*, dan *accuracy* pada *repository* metada didapatkan *recall* 92,1%, *precision* 87,7% dan *accuracy* 96,8%. Hasil ini didapatkan lebih baik dari pada *repository* data yaitu *recall* 35%, *precision* 35% dan *accuracy* 100% (Astrininditya, 2017).

Penelitian ini menggunakan *Resource Description Framework* (RDF), untuk membuat *metadata* dari dokumen XPDL.RDF adalah infrastruktur yang memungkinkan pengkodean, pertukaran, dan penggunaan kembali *metadata* terstruktur. Infrastruktur ini memungkinkan interoperabilitas *metadata* melalui desain mekanisme yang mendukung konvensi umum semantik, sintaksis, dan struktur (Miller, 2005). Penelitian ini membagi ruang lingkupnya menjadi data *workflow* dan *metadata workflow*. Data *workflow* adalah dokumen XPDL yang sebenarnya dan *metadata workflow* adalah keseluruhan informasi dari data *workflow* sebenarnya yang ditulis dengan kerangka kerja RDF. Rancangan sistem pada penelitian ini dimulai dari pengumpulan data *workflow* dari hasil observasi peneliti bersangkutan dan menyimpan datanya pada sistem secara manual. Setelahnya, *query* data tersebut menggunakan SQL untuk diolah dalam proses *parsing*. Selanjutnya, *parsing* XPDL untuk mengambil *metadata* yang diperlukan dan menyusunnya dalam sebuah dokumen RDF. Selanjutnya dilakukan interpretasi (usaha menemukan makna) *metadata* dan memvalidasi kebenaran RDF. Setelah validasi dilakukan, RDF disimpan dalam

repository. Tahap terakhir, *query* metadata menggunakan SPARQL untuk mengambil *workflow* yang diinginkan *user*.

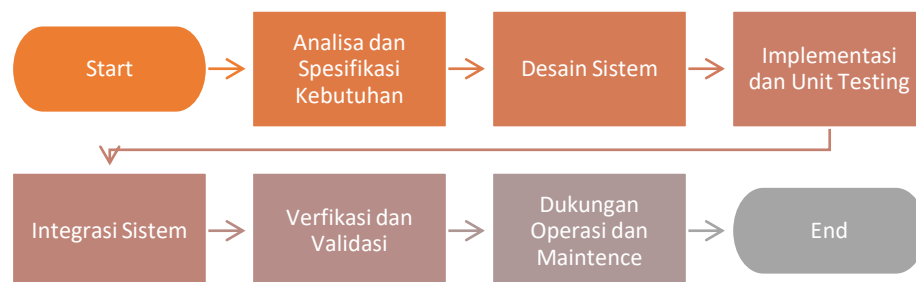
Proses penentuan *workflow* hasil dari pencarian pada penelitian ini berdasarkan pada penilaian nilai *recall*, *precision* dan *accuracy*. *Recall* adalah proporsi jumlah dokumen yang ditemukan kembali oleh pencarian di sistem *information retrieval*. *Precision* adalah kecocokan antara permintaan pencarian informasi dan hasil terhadap permintaan. Pencocokan pada penelitian ini menggunakan *exact string-matching*, yaitu pencocokan *string* secara tepat dengan menggunakan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang sama. Dari nilai *recall* dan *precision* akan didapatkan nilai *accuracy* yang merupakan tingkat kedekatan antara nilai prediksi dengan nilai sesungguhnya.

3. Komponen BPMS

Source code penelitian dengan judul Analisis dan Perancangan *Software* BPM Sistem. Penelitian ini masih dalam pengerjaan sampai saat ini. Pada proses merancang sistem BPM dilakukan dengan dua perancangan utama, yaitu arsitektur data dan arsitektur aplikasi. arsitektur data merupakan perancangan *database* sedangkan arsitektur aplikasi merupakan perancangan alur kerja serta tampilan aplikasi. sistem BPM yang akan dibangun, mempunyai tugas untuk manajemen *user*, *workflow*, *web service* dan *tenant*. Hasil dari penelitian ini adalah aplikasi BPMS untuk memudahkan pelaku usaha dalam mengubah model proses bisnisnya (Shomad Bukhori, 2020).

3.3 Metode

Metode penelitian yang akan digunakan sebagai pedoman alur pengembangan adalah *Component Based Development* menggunakan *V Model*. *V Model* digunakan dalam penelitian ini dikarenakan komponen telah siap dan komponen terpisah satu sama lain. Pengembangan menggunakan *V Model* berfokus pada pengembangan dan pemilihan komponen sehingga dapat digunakan untuk pedoman alur penelitian ini. Bagan alur proses pengembangan dengan *V Model* dapat dilihat pada gambar 3.1 di bawah.



Gambar 3. 1 Prosedur penelitian

3.3.1 Analisis dan Spesifikasi Kebutuhan

Tahap ini merupakan tahap menganalisis dan merealisasikan solusi terkait pada tahap selanjutnya untuk memenuhi kebutuhan yang diperlukan. Tahap ini penting untuk mengetahui komponen bisa diimplementasikan seutuhnya atau memerlukan modifikasi dulu sebelumnya atau menggunakan komponen baru karena bersifat *long-time support*.

Analisis komponen dilakukan pada dua bagian setiap komponen. Bagian pertama adalah analisis pada lingkup arsitektur komponen. Bagian kedua adalah analisis pada lingkup *source code* arsitektur. Spesifikasi kebutuhan ditentukan setelah melakukan analisis pada setiap komponen. Hasil dari proses spesifikasi

kebutuhan adalah kebutuhan fungsional dan non fungsional dari pengembangan sistem.

3.3.2 Desain Sistem

Tahap ini berhubungan dengan kemampuan atau fleksibilitas setiap komponen untuk diintegrasikan. Tahap ini menganalisis potensi suatu komponen konflik dengan komponen yang lain. Tahap ini akan membuat arsitektur baru untuk *software* yang dikembangkan sesuai dengan analisis yang telah dilakukan. Hasil desain sistem adalah poin-poin standarisasi sehingga komponen terpisah dapat diintegrasikan dan berjalan dengan baik.

3.3.3 Implementasi dan *Unit Testing*

Secara umum, metode yang paling ideal dalam pengembangan *software* CBD adalah menggunakan integrasi secara langsung. Komponen pada metode ini seharusnya sudah final dan teruji. Meskipun penggabungan secara metode ini memungkinkan, penggabungan ini masih harus diuji sampai penggabungan semua komponen berhasil. Tahap ini akan melakukan pengujian dengan *input* beberapa *query* sesuai dengan penelitian sebelumnya lalu *output* yang dihasilkan akan dicatat. *Input* dan *output* juga akan dievaluasi secara *source code* dan disesuaikan dengan *input* dan *output* pada komponen sistem BPM.

3.3.4 Integrasikan Sistem

Tahap integrasi sistem ini akan melakukan penggabungan setiap komponen menjadi satu kesatuan. *Framework* yang membangun pada setiap komponen juga dimasukkan dan diintegrasikan ke sistem utama (komponen yang menjadi *environment* utama). Tahap ini akan melakukan point-point yang telah disebutkan pada tahap desain sistem.

3.3.5 Verifikasi dan Validasi Sistem

Pengujian dan verifikasi standarisasi dilakukan pada tahap ini dengan tujuan untuk menemukan adanya *error*. Tahap ini memainkan peran penting karena struktur data *input* dan *output* harus sesuai rancangan sehingga *software* dapat berjalan. Tahap ini dilakukan *testing* pada setiap komponen yang telah diintegrasikan dengan menggunakan *query* yang sama pada tahap unit *testing*.

3.3.6 Dukungan dan Operasi Perawatan

Tahap sebelumnya telah memvalidasi kelancaran sebuah *software* dan siap untuk *release*. Tetapi tidak menutup kemungkinan, bahwa suatu saat terdapat masalah yang akan terjadi sehingga memerlukan perawatan dan dukungan operasi. Tahap ini tidak dilakukan dalam proses ini, karena *software* pada tahap ini masih belum beroperasi dan belum digunakan secara normal.

3.4 Skenario Pengujian

Skenario pengujian akan dilakukan untuk menjawab rumusan masalah yang diajukan. Skenario pengujian pada penelitian ini akan dilakukan perhitungan kompleksitas struktur *workflow* dan perhitungan *accuracy*. Perhitungan kompleksitas akan menggunakan persamaan The Yaqin Complexity (Yaqin et al., 2020) yang akan dijelaskan pada sub-bab di bawah. Perhitungan kompleksitas struktur *workflow* selanjutnya akan dibandingkan antara sistem BPM sebelum diintegrasikan dan sesudah integrasi. Perbandingan ini juga menghitung jumlah klik yang dibutuhkan *user* untuk konfigurasi. Pengujian bertujuan untuk menjawab rumusan masalah terhadap kecepatan waktu konfigurasi yang diperlukan dengan kompleksitas yang tinggi.

Skenario pengujian kedua adalah perhitungan akurasi terhadap *web service discovery* menggunakan persamaan *accuracy* (Wang et al., 2010). Persamaan *accuracy* lebih rinci akan dijelaskan pada sub-bab di bawah. Perhitungan akurasi juga akan dilakukan pada sistem BPM sebelum dan sesudah integrasi. Perhitungan ini tujuannya adalah menjawab rumusan masalah terhadap nilai *accuracy web service discovery* sehingga dapat memangkas waktu *user* untuk konfigurasi

3.4.1 The Yaqin Complexity

The Yaqin Complexity telah divalidasi dengan kriteria Weyuker's dimana dari sembilan kriteria, delapan kriteria telah terpenuhi. Weyuker mengusulkan sembilan kriteria untuk mengevaluasi perhitungan kompleksitas *software* pada saat pemrograman procedural dominan (Misra, 2009). Dalam jurnal tersebut disebutkan bahwa kompleksitas sebuah bisnis proses dapat diukur melalui metrik dalam representasi diagram atau bahasa standar. Pengukuran kompleksitas model proses bisnis dapat menggunakan pendekatan berdasarkan teori rekayasa perangkat lunak, ilmu kognitif, dan graph.

$$Ns = Ss + Es + Is + Acs + Bt$$

Dimana :

- Ns : Number of Nodes
- Ss : Starts Size
- Es : Ends Size
- Is : Intermediates Size
- Acs : Activities Size
- Bt : Number of *branching* types

$$Bt = ANDs + XORs + ORs$$

Dimana :

- $ANDs$: number of AND *branching*

- $XORs$: number of XOR *branching*
- ORs : number of OR *branching*

$$C_{AND} = 4. \sum_i (n!)_i$$

Dimana :

- C_{AND} : AND Branch Complexity
- n : number of *branches* in each *branch*
- i : index of AND *branches* in the business process model

$$C_{XOR} = 3. \sum_i (n!)_i$$

Dimana :

- C_{XOR} : XOR Branch Complexity

$$C_{OR} = 7. \sum_i \left(\sum_{k=1}^{n=1} \frac{n!}{(n-k)!} \right)_i$$

Dimana :

- C_{OR} : OR Branch Complexity
- k : the number of *branches* that may be passed at a *branch*
- i : index of OR *branches* in the business process model

$$C_{cyc} = 3. \frac{Acs_{cyc}}{Dm}$$

Dimana :

- C_{cyc} : Cyclic Complexity
- Acs_{cyc} : number of activities involved in a loop
- Dm : Diameter

$$D_{avg} = \frac{\sum_j^{Acs} D_j}{Acs}$$

Dimana :

- D_{avg} : average depth of each activity
- j : number of activities
- D : Depth of each activities

$$C_D = 14. D_{avg}$$

Dimana :

- C_D : complexity of depth

$$YC = NS + AS + C_{AND} + C_{XOR} + C_{OR} + C_{cyc} + C_D \quad (1)$$

dimana :

- YC : Yaqin Complexity

3.4.2 Accuracy

Accuracy mengukur seberapa benar tes diagnostik mengidentifikasi dan mengecualikan kondisi tertentu. Nilai *accuracy* mewakili proporsi hasil positif benar (TP dan TN) dalam populasi yang dipilih (Wang et al., 2010).

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Keterangan :

(TP + TN) : *Number of correct assessments*

(TP + TN + FP + FN) : *Number of all assessments*

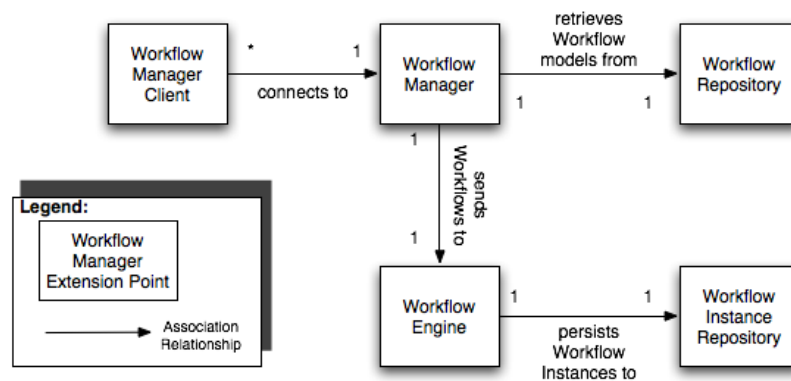
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengembangan *Software*

4.4.1 Spesifikasi dan *Analisis Kebutuhan*

Analisis yang dilakukan adalah analisis arsitektur dan analisis *source code* di setiap komponen. Analisis arsitektur melakukan analisis pada arsitektur program komponen. Analisis ini memiliki tujuan yaitu mendapatkan gambaran awal dari sebuah program dan pada bagian mana program dapat diimplemanastikan.



Gambar 4. 1 Arsitektur komponen *workflow repository*

Analisis pertama adalah terhadap komponen *workflow repository*. Program ini memiliki dua bagian untuk penyimpanan yaitu *Workflow Repository* dan *Workflow Instance Repository*. Bagian *workflow repository* digunakan untuk menyimpan file asli hasil *upload XPD*L oleh *client*. Bagian *workflow instance repository* digunakan untuk menyimpan metadata dari hasil proses *parsing XPD*L oleh *client*. Jadi, setiap *XPD*L yang diunggah oleh *client* akan di proses untuk diambil *metadata*-nya yang nantinya akan berguna untuk proses pencarian *workflow*. *Metadata* yang diambil akan disimpan dengan *framework* dan format *RDF*. *Metadata* yang diambil pada tahap *parsing* ini adalah sebagai berikut :

1. *XPDL Version*
2. *Vendor*
3. *Created*, tanggal dibuat.
4. *ModificationDate*
5. *Description*
6. *Author*
7. *CountryKey*
8. *Activity*, berisi nama *activity* atau proses pada file XPLD.

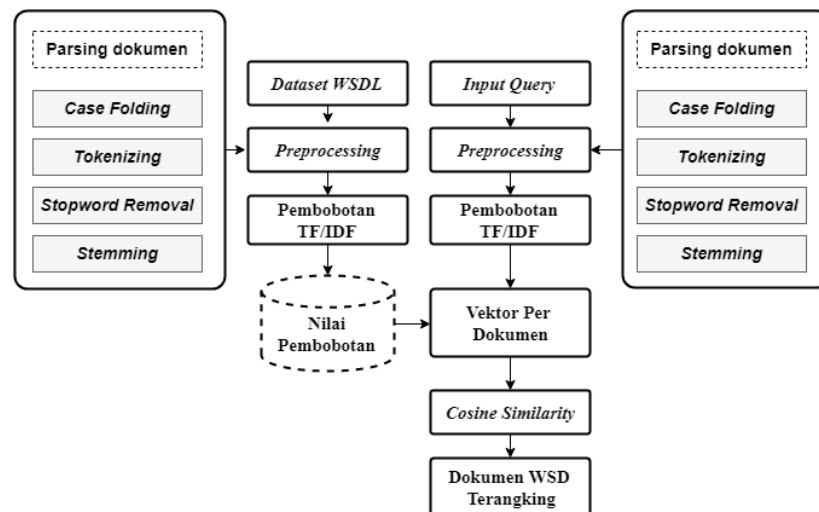
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
    <dc:URLData>xpd1-y07IlaqGx-appsxlntrTPjMF-20201111130319.xpd1</dc:URLData>
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler</dc:Vendor>
    <dc:Created>2017-12-03 09:53:53</dc:Created>
    <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
    <dc:Description>Akademik Pondok Pesantren Anwarul Huda</dc:Description>
    <dc:Author>digbys</dc:Author>
    <dc:Version>12</dc:Version>
    <dc:CountryKey></dc:CountryKey>
    <dc:Activity>| | Daftar ulang santri lama| Daftar Ulang santri baru| Pembuatan Kegiatan Pondok| Pendataan Santri pondok| Rekap pendataan kegiatan Di
  </rdf:Description>
</rdf:RDF>
```

Gambar 4. 2 XML dari RDF

Analisis kedua yaitu terhadap komponen *Web Service Discovery*. Arsitektur komponen *web service discovery* menggunakan *cosine similarity* dapat dilihat pada gambar 4.3. Sesuai dari gambar, terdapat dua proses utama pada komponen ini. Proses pertama pembobotan untuk kata kunci *web service repository* dan pembobotan pada *input query* pencari terhadap *web service repository*. Kedua proses tersebut dilakukan pembobotan menggunakan TF/IDF. Pembobotan kata kunci *web service repository* dilakukan saat mendaftarkan *web service* baru pada *web service repository*. Tahap ini, *client* harus mendefinisikan beberapa kata kunci yang sesuai dengan *web service* tersebut. Setelah dilakukan pembobotan saat *input web service* baru, bobot akan disimpan dan nilai bobot *web service* lama juga

berubah ketika terdapat kata kunci *web service* baru yang sama dengan *web service* lama.

Proses pencarian pada *web service discovery* ini menggunakan metode *cosine similarity*. Proses alur pencarian yang pertama adalah dilakukan *input query* kata kunci yang akan dicari. Setelah *query* dilakukan pembobotan, akan dibandingkan dengan hasil pembobotan terhadap *web service repository*. Proses selanjutnya akan dicari nilai dari vektor perdokumen dan dihitung nilai *cosine similarity*-nya. Hasil dari perhitungan *cosine similarity* adalah dokumen WSD terangking yang sesuai dengan *query* dimasukkan.

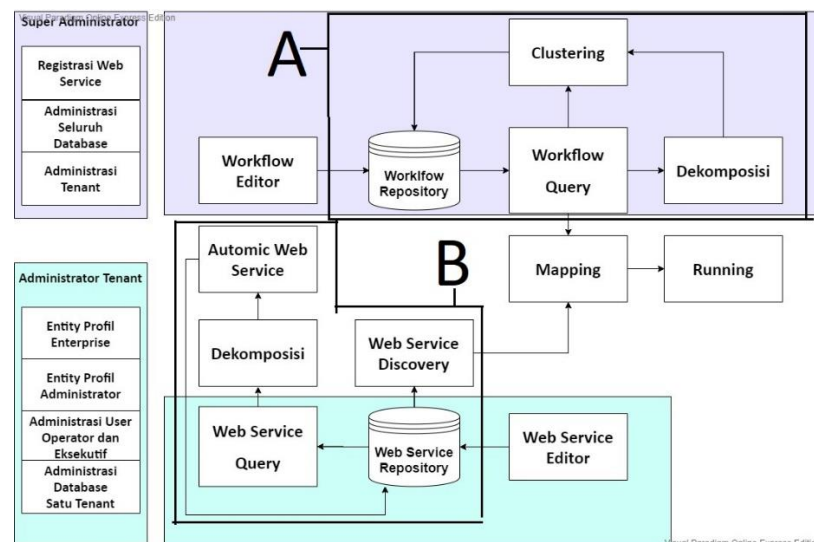


Gambar 4. 3 Arsitektur komponen *web service discovery*

Analisis ketiga yaitu analisis terhadap komponen sistem BPM. Arsitektur sistem BPM dapat dilihat pada gambar 4.4. Arsitektur sistem BPM pada penelitian ini terdapat pengecualian oleh peneliti yang bersangkutan. Pada gambar 4.4, terdapat proses *composing BPEL* setelah proses *web service discovery*. Proses ini dihilangkan pada penelitian ini. File BPEL pada proses BPM secara umum, digunakan untuk konfigurasi *running* sehingga dapat dijalankan layaknya sebuah program. Peneliti bersangkutan akan menggunakan *enviromtent* sendiri untuk

running pada penelitiannya. Koneksi antar *web service* dan proses *workflow* dihubungkan secara *database* pada penelitiannya. Hal ini dilakukan oleh peneliti yang bersangkutan dengan alasan tidak terlalu menggunakan *third-party* terlalu banyak sehingga peneliti bisa melakukan konfigurasi secara penuh.

sistem BPM ini memiliki dua akses *level*. Level pertama yaitu superadmin dan kedua yaitu *tenant*. Superadmin bertugas memanajemen *tenant* yang terdaftar dan manajemen *web service*. Data *web service* pada sistem BPM ini bersifat umum, yang berarti *web service* terdaftar dapat digunakan oleh semua *tenant*. *Tenant* berperan sebagai pengguna layanan. *Tenant* dapat diibaratkan dengan seseorang yang ingin menjalankan aplikasi hasil dari dia membuat *workflow* dengan ekstensi. XPD. *Tenant* dapat mengunggah file XPD. dan menjalankannya dengan *web service* yang terdaftar di *level* superadmin. File XPD. yang diunggah oleh setiap *tenant* bersifat *private*, yang berarti XPD. diunggah hanya dapat digunakan oleh *tenant* bersangkutan.



Gambar 4. 4 Arsitektur komponen sistem BPM

Analisis terhadap *source code* dilakukan setelah melakukan analisis arsitektur setiap komponen. Hal ini dilakukan untuk mendapatkan gambaran teknis bagaimana komponen diintegrasikan. Tujuan lain yaitu mendapatkan gambaran, struktur dan *environment source code* setiap komponen. Hasil dari proses analisis *sourc code* dijelaskan sesuai tabel berikut :

Tabel 4. 1 Hasil analisis *source code* komponen

Analisis	WF	WS	sistem BPM
Versi PHP	<5.x	5.x	7.x
<i>Framework</i>	Tidak	Tidak	Codeigniter 3
<i>OOP Ready</i>	Belum	Sudah	Sudah
Versi <i>Bootstrap</i>	3.3.7	3.3.7	4.4.1
<i>Third-Party</i>	RDF API V0.9.6 (Release 2008)	Tidak	Tidak

4.4.2 Desain Sistem

Desain sistem dan *software* berfungsi untuk menentukan integrase setiap komponen secara teknis. Penentuan ini didapatkan setelah melakukan analisis komponen dan menentukan spesifikasi kebutuhan. Arsitektur dari hasil integrasi selanjutnya telah dijelaskan pada gambar 3.7. *Worfkow repository* menggunakan *ontology* akan menggantikan *workflow repository* pada sistem BPM. *Workflow repository* pada sistem BPM hanya menyimpan file hasil *upload* tanpa menyimpan metadata *workflow*. *Web service discovery* menggunkana *cosine similarity* akan menggantikan proses *mapping web service* manual dengan menampilkan beberapa rekomendasi *web service* yang terkait.

Setelah melakukan analisis di tahap sebelumnya, dihasilkan beberapa ketentuan integrase setiap komponen. Beberapa ketentuan tersebut akan dijelaskan sebagai berikut :

1. Versi PHP yang digunakan adalah PHP versi 5.6. Hal ini dikarenakan PHP versi 5.6 adalah versi peralihan dari versi 4 ke versi 7, sehingga beberapa fungsi PHP versi 4 yang hilang di PHP versi 7 masih diakomodir di PHP versi 5.
2. *Base framework* yang digunakan mengikuti *framework* komponen sistem BPM.
3. Komponen yang belum *OOP ready* harus dikoding dengan format *OOP ready*.
4. *Third party* akan menjadi pada setiap komponen akan dimasukan ke base *framework* menjadi *library*.
5. *User interface* akan menggunakan komponen milik sistem BPM. Hal ini dikarenakan 2 komponen yang lain bersifat menyempurnakan komponen sistem BPM.
6. Database akan digabung menjadi satu dari ketiga komponen. Penggabungan *database* juga memerlukan beberapa *routing* pada tabel yang akan berperan untuk integrasi.
7. *Input* dan *output* akan ditentukan pada tahap unit *testing*.

4.4.3 Implementasi dan Unit Testing

Tahap ini akan dilakukan pengujian pada setiap komponen. Komponen akan diuji sesuai dengan data uji lama. Setiap pengujian akan dicatat. Pencatatan dilakukan sebagai tolak ukur pengujian komponen selanjutnya setelah tahap integrasi. Hasil yang diperoleh pada pengujian unit *testing* harus sesuai dengan pengujian komponen setelah integrasi.

Pengujian pertama dilakukan terhadap komponen *workflow repository*. Pengujian ini dilakukan dengan *input* beberapa *query* dan akan dicatat jumlah *output* yang dihasilkan. Pencarian komponen ini dilakukan pada metadata XPDL yang telah di-*parsing*. Pencarian tersebut menggunakan *query* SPARQL. Hasil yang diperoleh unit *testing* ini dijabarkan pada tabel di bawah. *Input query* pada komponen berbentuk data *string* dan *output* berbentuk data *array*.

Tabel 4. 2 Hasil unit *testing workflow repository*

<i>Query</i>	<i>Jumlah Output</i>
Tambah	4
Proses Tambah	4
Pilih Properti	4

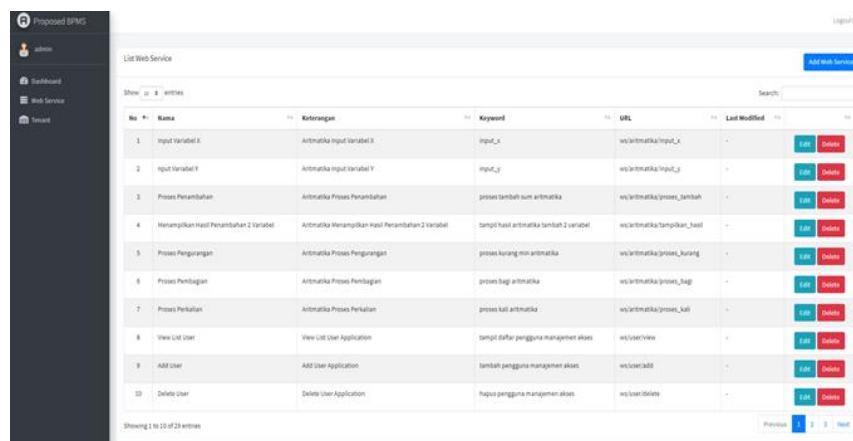
Pengujian kedua dilakukan terhadap komponen *web service discovery*. Pengujian akan dilakukan sama dengan draft laporan. Pengujian dilakukan dengan *input* beberapa *query* dan dihasilkan *output* beberapa kecocokan *web service* tersedia terhadap *query* pencarian. *Query input* akan dihitung bobot TF/IDF-nya dan akan dilakukan perhitungan vector pada setiap dokumen (hasil preprocessing dari *query input*) terhadap bobot TF/IDF dari *repository web service*. Setelahnya akan dihitung nilai *cosine similarity* terhadap nilai vektor yang dihasilkan. Proses ini berjalan di *source code* dengan nama fungsi *keyword()* dan *panggilcache()*. *Input* dari *query* ini berbentuk tipe data *string* dan *output* dihasilkan berbentuk tipe data *array*. Kolom ini merupakan acuan keberhasilan untuk proses selanjutnya, apakah setelah proses integrasi mengembalikan nilai yang sama atau tidak.

Tabel 4. 3 Hasil unit *testing web service discovery*

<i>Query</i>	<i>Jumlah Output</i>
Properti	8
Manajemen	4

Bayar	8
-------	---

Pengujian ketiga dilakukan terhadap komponen sistem BPM. Pengujian ini dilakukan di *level* akses superadmin dan *tenant*. Pengujian superadmin berfokus pada manajemen *web service*. Pengujian *tenant* akan berfokus pada proses *upload* file XPDL sampai pemetaan *web service*. Pengujian manajemen *web service* dilakukan secara *blackbox testing*. Hasil dari pengujian, hanya terdapat fitur CRUD tanpa ada proses lagi di dalamnya.



No	Nama	Keterangan	Keyword	URL	Last Modified
1	Input Variabel X	Automatic Input Variabel X	input_x	ws/automatic/input_x	-
2	Input Variabel Y	Automatic Input Variabel Y	input_y	ws/automatic/input_y	-
3	Proses Penambahan	Automatic Proses Penambahan	proses tambah sum automatic	ws/automatic/proses_tambah	-
4	Mengambil Hasil Penambahan 2 variabel	Automatic Mengambil Hasil Penambahan 2 variabel	ambil hasil automatic tambah 2 variabel	ws/automatic/ambil_hasil_tambah_2_variabel	-
5	Proses Pengurangan	Automatic Proses Pengurangan	proses kurang min automatic	ws/automatic/proses_kurang	-
6	Proses Pembagian	Automatic Proses Pembagian	proses bagi automatic	ws/automatic/proses_bagi	-
7	Proses Perkalian	Automatic Proses Perkalian	proses kali automatic	ws/automatic/proses_kali	-
8	View List User	View List User Application	ambil daftar pengguna manajemen akses	ws/user/view	-
9	Add User	Add User Application	tambah pengguna manajemen akses	ws/user/add	-
10	Delete User	Delete User Application	hapus pengguna manajemen akses	ws/user/delete	-

Gambar 4. 5 GUI manajemen *web service*

Pengujian selanjutnya dilakukan pada *level tenant*. Pertama, akan dilakukan pengujian *upload* dan melihat proses yang terjadi di dalamnya. Setelah melakukan percobaan, setelah melakukan *trigger* untuk proses *upload*, terdapat pemetaan pada proses *workflow* pada file XPDL dan nama pada tag *line*-nya. Tag *line* akan diidentifikasi sebagai *user level*. Pemetaan akan disimpan di *database*. Proses pemetaan *workflow* menggunakan tag *to* dan *from* sehingga bisa di ambil alur *workflow* sesuai pada saat desain. Proses selanjutnya adalah melakukan pemetaan *web service*. Pemetaan dilakukan secara manual dengan cara merelasikan

relasi antara proses *workflow* dan *web service* tersedia. Setelah konfigurasi pemetaan setiap proses *workflow* selesai, file XPDL sudah dapat di-*running*.

Gambar 4. 6 GUI *upload* XPDL

Nama Proses	Role	Web Service
START	-	-
Input x	Pilih Role	Pilih Web Service
Input y	Pilih Role	Pilih Web Service
OR GATEWAY (SPLIT)	-	-
Proses Tambah	Pilih Role	Pilih Web Service
Proses Tambah	Pilih Role	Pilih Web Service
Proses Tambah	Pilih Role	Pilih Web Service
Proses Tambah	Pilih Role	Pilih Web Service
OR GATEWAY (JOIN)	-	-
Tampilkan Hasil	Pilih Role	Pilih Web Service
END	-	-

Gambar 4. 7 GUI pemetaan *web service*

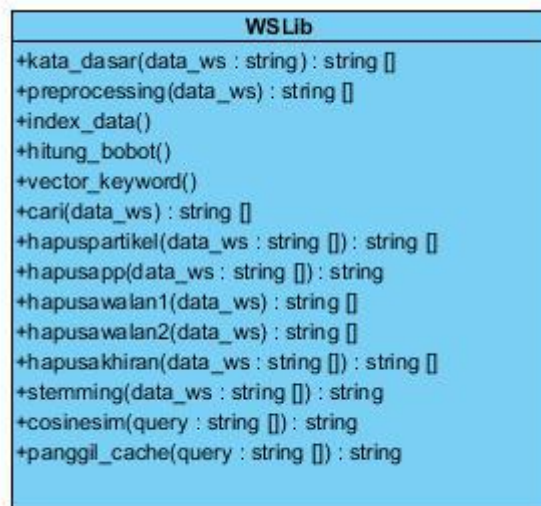
4.4.4 Integrasi Sistem

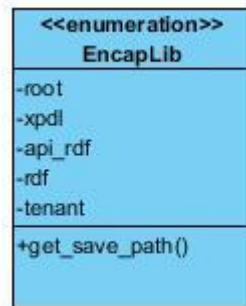
Tahap ini akan mengerjakan point-point yang telah disebutkan pada tahap desain sistem. Pengerjaan pertama yaitu penggabungan dan normalisasi *database*. Di sini juga dilakukan *routing* pada tabel yang digunakan menghubungkan ketiga komponen. Berikut disebutkan beberapa tabel yang berperan penting dalam *routing*:

Tabel 4. 4 Tabel *routing database*

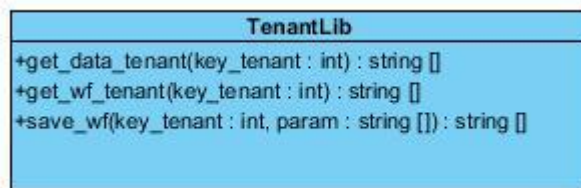
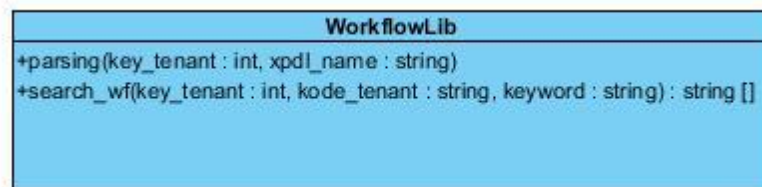
No	Nama Tabel Routing	Keterangan
1	tb_web_service	Menggantikan tabel <i>web service</i> lama dengan tabel komponen sistem BPM
2	tb_repo_data_n	Mengubungkan tabel komponen <i>workflow repository</i> dengan tabel <i>workflow repository</i> sistem BPM
3	tb_tenant	Menghubungkan konfigurasi <i>tenant</i> sistem BPM dengan komponen <i>workflow repository</i>

Pengerjaan kedua yaitu pengkodean ulang pada *workflow repository* karena belum *OOP Ready*. Pada proses ini, dibuat dua fungsi penting, yaitu *parsing()* dan *search_wf()*. Fungsi *parsing()* berguna untuk mengambil *metadata* file XPDL dan disimpan dalam format *rdf*. Fungsi *search_wf()* berguna untuk pencarian *workflow repository*. Pemetaan *class* pada *source code* dapat dilihat dari UML di bawah.

Gambar 4. 8 UML Class *WorkflowLib*



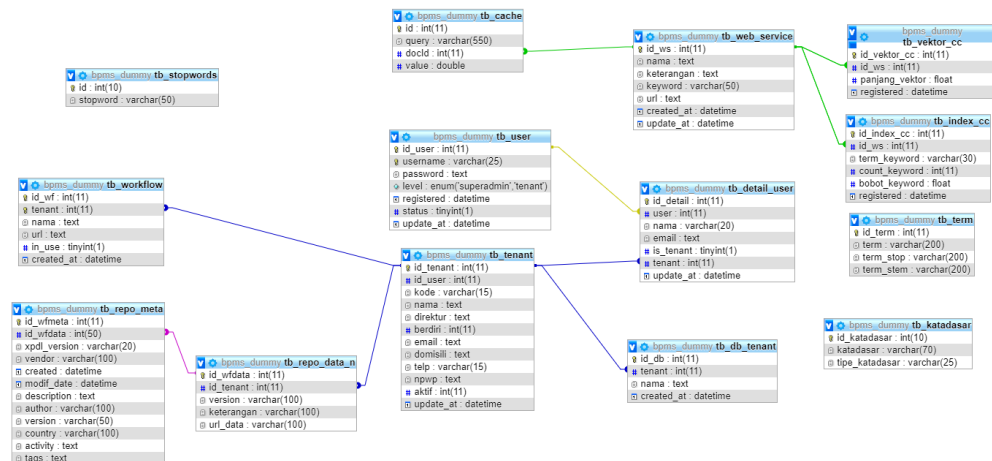
Gambar 4. 9 UML Class EncapLib

Gambar 4. 10 UML class *TenantLib*

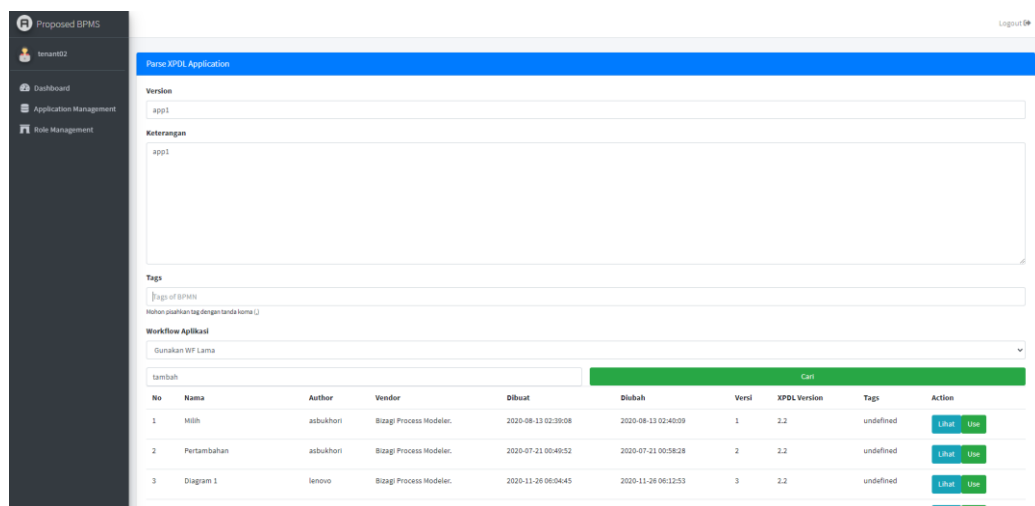
Gambar 4. 11 UML class WSLib

Pada pengerjaan tahap ini, juga terdapat masalah yang tidak terdeteksi di tahap desain sistem. Masalah yang terjadi adalah konflik antara *third-party workflow repository* yaitu RDF for PHP (RAP) V0.9.6 dan *framework* Codeigniter 3 pada komponen sistem BPM. Secara teknis, adalah masalah terjadi akibat Codeigniter mengharuskan menggunakan fungsi yang didukung oleh veris PHP selanjutnya. Konflik fungsi yang terjadi adalah fungsi *ereg()* yang digantikan dengan fungsi

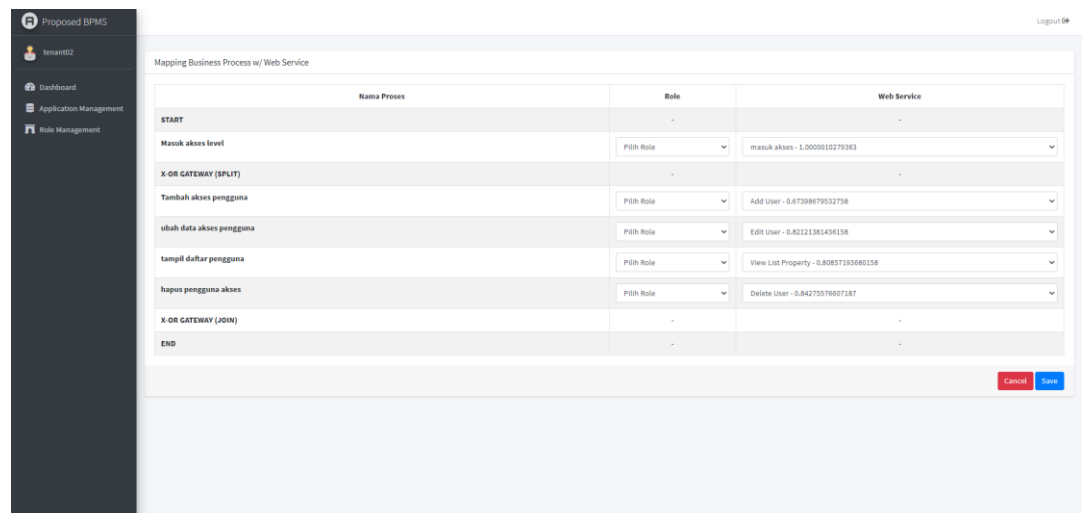
preg_match(). Solusi untuk masalah ini, membuat *environment* terpisah untuk *third-party* RAP dan dibuatkan API agar bisa dipanggil oleh komponen utama.



Gambar 4. 12 Database ketiga komponen



Gambar 4. 13 GUI integrasi *workflow repository*

Gambar 4. 14 GUI integrasi *web service discovery*

4.4.5 Verifikasi dan Validasi Sistem

Tahap ini dilakukan *testing* pada setiap komponen yang telah diintegrasikan dengan menggunakan *query* yang sama pada tahap unit *testing*. *Testing* juga dilakukan menggunakan data pengujian sama dengan tahap unit *testing*. Pengujian pada *workflow repository* dan *web service discovery* menghasilkan jumlah *output* yang sama pada setiap *query*-nya. Hasil pengujian validasi *workflow repository* tercantum pada Tabel 4.5. Hasil pengujian validasi *web service discovery* tercantum pada Tabel 4.6. Kedua pengujian tersebut memiliki nilai jumlah *output* dengan tahap implementasi dan unit *testing*, Tabel 4.2 dan Tabel 4.3.

Tabel 4. 5 Hasil validasi *workflow repository*

<i>Query</i>	<i>Jumlah Output</i>
Tambah	4
Proses Tambah	4
Pilih property	4

Tabel 4. 6 Hasil validasi *web service discovery*

<i>Query</i>	<i>Jumlah Output</i>
Properti	8
Manajemen	4
Bayar	8

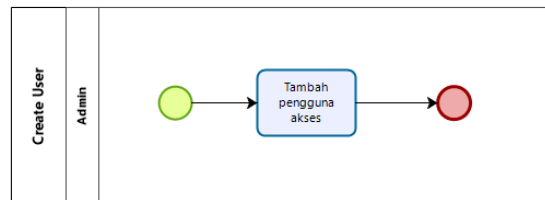
4.2 Hasil Eksperimen

4.2.1 Kompleksitas *Workflow*

Workflow data uji yang akan digunakan pengujian kompleks adalah *workflow* tambah *user*, penambahan properti baru dan manajemen *user*. Data primer *workflow* tidak digunakan semua karena *workflow* data uji ini karena jika digunakan semua akan menyebabkan penambahan data uji *web service discovery*. *Workflow* data uji yang digunakan ini akan digunakan juga selanjutnya untuk data uji *web service discovery*.

Workflow data uji yang adalah sebagai berikut :

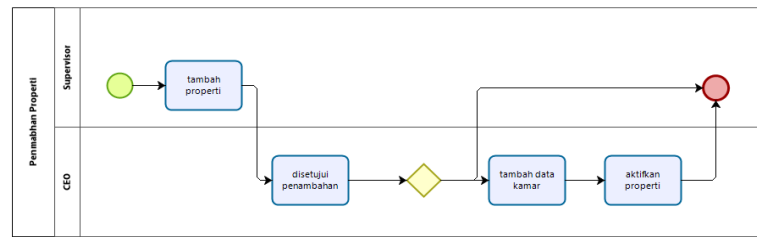
1. *Workflow* tambah *user*



Gambar 4. 15 *workflow* tambah *user*

Workflow ini merupakan *workflow* dengan *flow* yang tersusun secara *sequence*. *Workflow* tersebut terdiri dari 1 *activities*, 1 *start node* dan 1 *end node*.

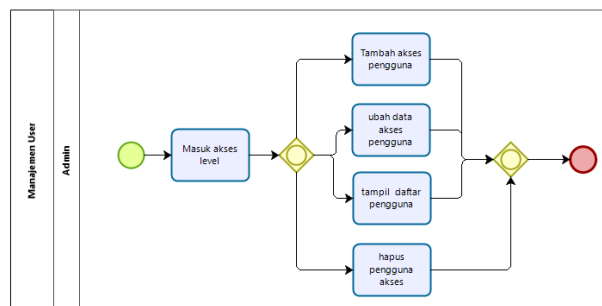
2. *Workflow* tambah properti baru



Gambar 4. 16. *workflow* tambah properti baru

Workflow tersebut menggunakan tipe percabangan *exclusive* dalam prosesnya. *Workflow* pada gambar di atas terdiri dari 4 *activities*, 1 *branch*, 1 *start node* dan 1 *end node*.

3. *Workflow* manajemen user



Gambar 4. 17. *workflow* manajemen user

Workflow gambar di atas menggunakan tipe percabangan *inclusive* dalam prosesnya. *Workflow* tersebut terdiri dari 5 *activities*, 2 *branch*, 1 *start node* dan 1 *end node*.

Pengujian tahap pertama adalah pengukuran komplestitas struktur *workflow* bisnis proses. Perhitungan kali ini akan dicontohkan dengan *workflow* gambar 2. Perhitungan dilakukan berurutan sesuai step dicantumkan di atas.

Pertama, akan dilakukan perhitungan N_s menggunakan persamaan 1 dan 2 dan didapatkan hasil sebagai berikut :

$$\text{Step 1 : } Ns = 1 + 1 + 0 + 4 + (0 + 1 + 0) = 6$$

karena *workflow* hanya memiliki percabangan exclusive (XOR), maka yang perhitungan kompleksitas percabangan hanya menggunakan persamaan 4. Sedangkan, nilai kompleksitas percabangan C_{AND} , C_{XOR} dan C_{OR} bernilai 0. Begitu juga, pada *workflow* tidak terdapat perulangan, sehingga nilai C_{cyc} bernilai 0. Perhitungan selanjutnya yaitu menghitung nilai kompleksitas ke dalam menggunakan persamaan 7 dan 8.

$$\text{Step 2 : } C_{XOR} = 3. (2!) = 6$$

$$\text{Step 3 : } Davg = \frac{6}{4} = 1,5$$

$$\text{Step 4 : } C_D = 14. 1,5 = 21$$

Seluruh nilai telah didapatkan sampai perhitungan ini. Hasil nilai YC selanjutnya dapat dihitung dengan menggunakan persamaan 9.

$$YC = 6 + 7 + 0 + 6 + 0 + 0 + 21 = 46$$

Sehingga perhitungan untuk seluruh *workflow* pada data uji adalah sebagai berikut :'

Tabel 4. 7 Perhitungan The Yaqin Complexity

No	Workflow	Ns	As	C _{AND}	C _{XOR}	C _{OR}	C _{cyc}	C _D	YC
1	tambah <i>user</i>	3	2	0	0	0	0	0	5
2	Tambah property baru	6	7	0	6	0	0	21	46
3	Manajemen <i>user</i>	9	11	0	0	168	0	25,2	213,2

Pengujian terhadap *workflow repository* menghasilkan nilai kompleksitas berebeda-beda. Hasil pengujian pada tabel di atas, BPMS dapat menjalankan *workflow* dengan nilai kompleksitas YC dengan nilai 213,2. Nilai kompleksitas berbanding lurus dengan banyaknya elemen yang ditambahkan. Hal ini sesuai dengan persamaan YC, yang persamaannya menggunakan operator aritmatika pertambahan. Dari hasil perhitungan ketiga *workflow*, BPMS dapat menjalankan kompleksitas dengan nilai cukup tinggi. Kompleksitas tidak berpengaruh dengan performa *workflow repository* sehingga dapat menerima *workflow* dengan struktur kompleks. *Workflow repository* dapat membaca struktur kompleks, karena *parsing* dilakukan pada nama proses *workflow*. Hal ini menyebabkan penyimpanan *workflow* lebih tertata dan dapat digunakan kembali ketika akan digunakan. Karena perhitungan kompleksitas dilakukan pada *workflow*, maka nilai kompleksitas sebelum dan sesudah integrasi memiliki nilai yang sama.

Pengaruh kompleksitas terhadap waktu konfigurasi *workflow* dapat dihitung dari jumlah klik yang diperlukan untuk konfigurasi. Semakin sedikit jumlah klik menandakan waktu yang digunakan untuk konfigurasi juga semakin sedikit. Tabel 4.8 menunjukkan perbandingan jumlah klik sebelum dan sesudah integrasi.

Tabel 4. 8 Perbandingan jumlah klik

Nama Workflow	YC	Jumlah Klik			
		Sebelum		Sesudah	
		WF	WS	WF	WS
Tambah user	5	4	5	3	3
Tambah property baru	46	4	13	3	9
Manajemen user	213,2	4	21	3	13
Jumlah		12	39	9	25

	51	34
--	----	----

Tabel 4. 9 Rasio perbandingan dan peningkatan

Nama Workflow	YC	Rasio Perbandingan				Peningkatan	
		Sebelum		Sesudah		WF	WS
		WF	WS	WF	WS		
Tambah <i>user</i>	5	0,8	1,0	0,6	0,6	0,2	0,4
Tambah property baru	46	0,09	0,28	0,07	0,2	0,02	0,09
Manajemen <i>user</i>	213,2	0,02	0,1	0,01	0,06	0,005	0,04

Hasil perbandingan jumlah klik sebelum dan sesudah integrasi menghasilkan jumlah klik sesudah integrasi lebih kecil dibanding sebelum integrasi. Hasil lebih kecil jumlah klik didapatkan pada kedua komponen integrasi. Jumlah klik *workflow repository* menghasilkan jumlah klik bernilai 12 untuk sebelum dan 9 untuk sesudah integrasi. Jumlah klik *web service discovery* menghasilkan jumlah klik sebelum yaitu 39 dan sesudah yaitu 25.

Tabel 4.9 menggambarkan rasio perbandingan antara kompleksitas *workflow* dengan jumlah klik yang digunakan untuk konfigurasi. Nilai rasio perbandingan didapatkan dengan membandingkan antara jumlah klik dengan nilai perhitungan kompleksitas *workflow*. Pada Tabel 4.9, terlihat nilai rasio perbandingan sesudah integrasi lebih kecil dibanding dengan sebelum integrasi. Nilai rasio perbandingan yang lebih kecil didapatkan pada kedua komponen integrasi. Pada *workflow* dengan nilai kompleksitas 5, didapatkan nilai rasio perbandingan *workflow repository* sebelum dan sesudah integrasi adalah 0,8 dan 0,6, sedangkan pada *web service discovery* adalah 1 dan 0,6. Pada *workflow* dengan nilai kompleksitas 46, didapatkan nilai rasio perbandingan *workflow repository* sebelum dan sesudah

integrasi adalah 0,09 dan 0,07, sedangkan pada *web service discovery* adalah 0,28 dan 0,2. Pada *workflow* dengan nilai kompleksitas 213,2, didapatkan nilai rasio perbandingan *workflow repository* sebelum dan sesudah integrasi adalah 0,02 dan 0,01, sedangkan pada *web service discovery* adalah 0,1 dan 0,06. Dari nilai rasio perbandingan dihasilkan juga nilai peningkatan pada setiap kompleksitas. Nilai peningkatan didapat dari selisih nilai rasio perbandingan antara sebelum dan sesudah integrasi. Pada nilai kompleksitas *workflow* 5, nilai peningkatan pada *workflow repository* adalah 0,2 serta nilai peningkatan pada *web service discovery* adalah 0,4. Pada nilai kompleksitas *workflow* 46, nilai peningkatan pada *workflow repository* adalah 0,02 serta nilai peningkatan pada *web service discovery* adalah 0,09. Pada nilai kompleksitas *workflow* 213,2, nilai peningkatan pada *workflow repository* adalah 0,005 serta nilai peningkatan pada *web service discovery* adalah 0,04.

4.2.2 Akurasi Web Service Discovery

Pengujian tahap kedua adalah pemetaan *web service discovery*. *Query* digunakan untuk pengujian adalah nama setiap proses *workflow*. Data uji *web service* berjumlah 51 *web service*. Penamaan nama proses terdiri 1 kata sampai 4 kata. Penamaan proses juga menggunakan kata dasar, kata dengan imbuhan, dan kata tidak baku. Pemetaan akan dihitung nilai *precision*, *recall* dan *accuracy*. Hasil perhitungan setiap *query* terinput ditampilkan pada tabel berikut :

Tabel 4. 10 Hasil hitung *accuracy web service discovery* sebelum integrasi

No	Query	TP	TN	FP	FN	P (%)	R (%)	A (%)	Urutan
1	masuk akses level	1	0	50	0	2	100	2	50
2	tambah akses pengguna	1	0	50	0	2	100	2	9
3	ubah data akses pengguna	1	0	50	0	2	100	2	11

4	tampil daftar pengguna	1	0	50	0	2	100	2	8
5	hapus pengguna akses	1	0	50	0	2	100	2	10
6	tambah properti	1	0	50	0	2	100	2	13
7	disetujui penambahan	1	0	50	0	2	100	2	30
8	tambah data kamar	1	0	50	0	2	100	2	49
9	aktifkan properti	1	0	50	0	2	100	2	47
10	tambah pengguna akses	1	0	50	0	2	100	2	31
Rata-rata						2	100	2	

Tabel 4. 11 Hasil hitung *accuracy web service discovery* setelah integrasi

No	Query	TP	TN	FP	FN	P (%)	R (%)	A (%)	Urutan
1	masuk akses level	1	46	4	0	20	100	92,2	1
2	tambah akses pengguna	1	46	4	0	20	100	92,2	1
3	ubah data akses pengguna	1	44	6	0	14,3	100	88,2	1
4	tampil daftar pengguna	1	47	3	0	25	100	94,1	4
5	hapus pengguna akses	1	45	5	0	16,7	100	90,2	1
6	tambah properti	1	43	7	0	12,5	100	86,3	2
7	disetujui penambahan	1	50	0	0	100	100	100	1
8	tambah data kamar	1	42	8	0	11,1	100	84,3	1
9	aktifkan properti	1	43	7		12,5	100	86,3	1
10	tambah pengguna akses	1	46	4	0	20,0	100	92,2	1
Rata-rata						26,2	100	91,07	

Pengujian pemetaan *web service discovery* untuk mencari *web service* yang sesuai memiliki nilai rata-rata *accuracy* 90,59%. Setiap pencarian menghasilkan

nilai *precision* yang rendah dengan rata-rata 25,2%. Berbanding terbalik dengan *precision*, nilai *recall* memiliki nilai rata-rata 100%. Hal ini menandakan dari setiap proses pemetaan *web service*, selalu mengembalikan sesuai jumlah *web service* yang benar.

Tabel 4.9 dengan nama kolom TP dan Urutan selalu memiliki nilai sama. Penulis ingin menunjukkan bahwa *web service* yang sesuai dengan pencarian berada di urutan nomor sekian pada hasil pencarian. Dari 10 kali pengujian, yang menghasilkan *web service* dicari pada urutan nomor 1 terdapat 8 kali dan bukan pada urutan nomor 1 terdapat 2 kali. *Query* yang tidak menghasilkan *web service* dicari adalah tampil daftar pengguna dan tambah properti. Hal ini dikarenakan terdapat beberapa kata kunci yang tidak terdeteksi pada saat tahap preprocessing. Dari 2 *query* tersebut, yang tidak terdeteksi adalah kata kunci tampil, pengguna dan tambah. Hal ini menyebabkan hasil pencarian menjadi tidak spesifik. Penyebab lain dari hasil tidak spesifik adalah penggunaan *query* atau nama proses yang kurang spesifik. Dari 10 pengujian, 2 pengujian menghasilkan urutan nomor 1 pada *web service* dicari, maka didapat nilai perbandingan 80 % untuk tingkat ketepatan.

Nilai yang dihasilkan oleh integrase *web service discovery* pada BPMS lebih baik dibandingkan BPMS sebelum integrase. Nilai *accuracy web service discovery* sebelum diintegrasikan ditampilkan pada tabel 4.10. Rata-rata nilai *accuracy* sebelum diintegrasikan adalah 2% meski memiliki nilai *recall* 100% pada setiap pengujiannya. Nomor urutan *web service* yang terkait dengan *query* dari pencarian memiliki nomor urut yang acak dan tidak selalu memiliki nomor urut di range 1 sampai 10. Hal ini dikarenakan, BPMS sebelum integrase hanya melakukan get data dari *database* tanpa dilakukan preproses untuk memudahkan pengguna melakukan

konfigurasi. Dari hasil ini dapat ditarik kesimpulan bahwa integrasi *web service discovery* membantu *user* untuk konfigurasi *workflow*, sehingga dapat memangkas waktu *user* untuk konfigurasi secara manual.

4.3 Pembahasan

Pengembangan *software* pada saat ini telah dilakukan dan diintegrasikan. Prosedur sesuai V Model pada *Component Based Development* telah dikerjakan dengan sesuai dan beraturan. Komponen-komponen kecerdasan telah diintegrasikan kepada sistem BPM. Implementasi dan integrasi dilakukan sesuai desain sistem dan didapatkan spesifikasi kebutuhan yang diinginkan. *Software* yang dikembangkan juga telah melewati verifikasi dan validasi sehingga *software* telah lolos uji.

Software telah melewati tahap skenario pengujian. Penelitian ini, mengajukan 2 skenario pengujian, yaitu pengujian pengukuran kompleksitas struktur *workflow* dan pengujian pengukuran akurasi *web service discovery*. Pengujian kompleksitas struktur *workflow* menggunakan persamaan *The Yaqin Complexity*. Pengujian akurasi *web service discovery* menggunakan persamaan *accuracy*. Hasil yang didapatkan dari perhitungan pengukuran kompleksitas memiliki nilai tertinggi 213,2. *Workflow* yang diuji bertipe *sequence workflow* dan *branching workflow*.

Paragraf penjelasan Tabel 4.8 dan 4.9 sebelumnya telah dijelaskan bagaimana mendapatkan nilai rasio perbandingan dan peningkatan. Hasil dari Tabel 4.9, didapatkan bahwa nilai perbandingan rasio selalu lebih kecil pada setiap kompleksitas dan. Nilai rasio perbandingan pada saat sebelum dan sesudah integrasi juga didapatkan, nilai rasio perbandingan sesudah lebih kecil dibanding dengan sebelum integrasi. Dari nilai tersebut, didapatkan juga nilai peningkatan yang semakin kecil pada *workflow repository* dan *web service discovery* di setiap

kompleksitas *workflow*. Dari percobaan ini dapat disimpulkan bahwa setelah integrasi, dapat mengurangi waktu konfigurasi dengan kompleksitas *workflow* yang ditandai lebih kecilnya nilai rasio perbandingan sehingga nilai peningkatan juga semakin kecil. Semakin kecil nilai peningkatan berarti waktu yang diperlukan untuk konfigurasi juga semakin sedikit.

Pengujian pengukuran akurasi telah dilakukan dan dibandingkan antara sistem BPM sebelum integrasi dan sistem BPM sesudah integrasi. Kedua hasil perhitungan memiliki nilai *recall* 100%, tetapi memiliki nilai *accuracy* berbeda. Nilai *accuracy* pada sistem BPM sebelum integrasi adalah 2%. Sedangkan nilai *accuracy* pada sistem BPM sesudah integrasi adalah 91,07%. Nilai *accuracy* sistem BPM sesudah integrasi memiliki nilai jauh yang lebih tinggi dibanding sistem BPM sebelum integrasi. Pengujian *web service discovery* pada sistem BPM sesudah integrasi memiliki hasil yang *web service*-nya terkait sesuai *query* di urutan nomor 1 sejumlah 8 percobaan dari 10 percobaan. Dua percobaan *web service discovery* yang lain untuk nomor urut *web service* terkait sesuai *query* berada di range antara 1 sampai 10 sehingga masih mempermudah *user* menemukan *web service* yang sesuai. konfigurasi. Dari hasil ini dapat ditarik kesimpulan bahwa integrasi *web service discovery* membantu *user* untuk konfigurasi *workflow*, sehingga dapat memangkas waktu *user* untuk konfigurasi secara manual.

4.4 Integrasi Sains dan Islam

Sains dan Islam tidak dapat dipisahkan dalam kehidupan sehari-hari. Islam menyuruh umatnya untuk berpegang teguh kepada Al-Qur'an yang merupakan pedoman hidup setiap umat muslim.. Terdapat salah satu ayat Al-Qur'an, yaitu surah Al-Isra ayat 19 yang berbunyi sebagai berikut :

وَمَنْ أَرَادَ الْآخِرَةَ وَسَعَىٰ لَهَا سَعْيَهَا وَهُوَ مُؤْمِنٌ فَأُولَٰئِكَ كَانَ سَعْيُهُمْ مَّشْكُورًا ۝١٩

Artinya : “Dan barangsiapa yang menghendaki kehidupan akhirat dan berusaha ke arah itu dengan sungguh-sungguh sedang ia adalah mukmin, maka mereka itu adalah orang-orang yang usahanya dibalasi dengan baik” (Q.S Al-Isra Ayat 19)

Setiap muslim dalam ajaran islam, disuruh untuk selalu menjadi lebih baik dari hari kemarin dan lebih baik esoknya dari hari ini. Aspek lebih baik bukan hanya saja sekedar dalam hal agama, tetapi semua aspek kehidupan seorang muslim. Hal ini termasuk juga dalam aspek pekerjaan atau lainnya.

Lebih baik merupakan anjuran bagi setiap muslim. Begitu juga dalam penelitian ini, penulis ingin membuat sistem BPM yang menjadi lebih baik lagi yaitu dengan menambahkan komponen *artificial intelligence* ke dalamnya. Introspeksi diri juga tidak lepas dalam usaha menjadi lebih baik. Meskipun sistem yang dibuat penulis telah menjadi lebih baik, selalu ada yang dapat dikembangkan untuk lebih baik lagi, semisal penambahan *artificial intelligence* lain ke dalam sistem BPM atau penggantian komponen dengan metode yang lebih baik lagi. *Wallahualam Bissawab.*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian dengan judul Pengembangan Sistem Pengelolaan Proses Bisnis Cerdas telah selesai dan sampai pada kesimpulan. Penelitian sudah sesuai dengan metode penelitian yang disusun. Pengembangan *software* sudah dibangun berdasarkan alur proses *V Model* pada *Component Based Development* dan sudah sesuai spesifikasi yang ditentukan pada tahap desain sistem. Pengembangan *software* juga sudah melalui tahap *unit testing* dan *system testing* dan didapatkan *software* dapat dijalankan dan dioperasikan.

Skenario pengujian terhadap *software* ada dua pengujian, yaitu perhitungan kompleksitas menggunakan persamaan The Yaqin Complexity dan pengujian perhitungan akurasi menggunakan persamaan *accuracy*. Pengujian dilakukan dengan maksud untuk mengetahui keterbaruan dengan penelitian sebelumnya. Berikut hasil yang didapatkan pengujian *software* :

1. Hasil dari rasio perbandingan dan peningkatan jumlah klik sebelum integrasi lebih besar dibanding sesudah integrasi dengan nilai YC yang telah dihitung sebelumnya pada setiap *workflow*. Nilai rasio perbandingan dan peningkatan tercantum di Tabel 4.9 sebelumnya. Dari nilai rasio perbandingan dan peningkatan tersebut, disimpulkan bahwa waktu yang diperlukan untuk konfigurasi juga semakin sedikit.
2. Hasil dari nilai *accuracy* sebelum integrasi jauh lebih kecil dibanding dengan nilai *accuracy* sesudah integrasi. Nilai *accuracy* sebelum integrasi adalah 2%, sedangkan nilai *accuracy* sesudah integrasi adalah 91,07%. Hal ini menunjukkan bahwa integrasi *web service discovery* membantu

untuk mengkonfigurasi *workflow*, sehingga dapat memangkas waktu untuk konfigurasi.

5.2 Saran

Peneliti memiliki saran untuk pengembangan sistem BPM selanjutnya agar mendapatkan hasil lebih baik sebagai berikut :

1. Sistem BPM dapat menjalankan *workflow* yang memiliki struktur percabangan bertipe *parallel*.
2. Sistem BPM dapat menjalankan *workflow* yang memiliki struktur bertipe *looping*.
3. Sistem BPM dapat dikembangkan lagi dengan menambahkan integrasi kecerdasan *interface repository* sehingga *interface* yang dibuat bersifat *reusable*

DAFTAR PUSAKA

- Ahmad, T., & Van Looy, A. (2020). Business Process Management and Digital Innovations: A Systematic Literature Review. *Sustainability*, 12(17), 6827. <https://doi.org/10.3390/su12176827>
- AL-Mahali, Jalaluddin, I., & As-Suyuti. (2007). *Tafsir Jalalain. Terj. Bahrn Abubakar*. Sinar Baru Algesindo.
- Astrininditya, R. (2017). Aplikasi *workflow repository* pada *enterprise resource planning* pondok pesantren menggunakan *resource description framework*. *Undergraduate Thesis, Universitas Islam Negeri Maulana Malik Ibrahim*. <http://etheses.uin-malang.ac.id/id/eprint/11704>
- Crnkovic, I., Larsson, S., & Chaudron, M. (2005). Component-based Development Process and Component Lifecycle. *IEEE Xplore*, 7. <https://doi.org/10.1109/ITI.2005.1491195>
- Dewi, R. M. (2019). *Web service discovery* menggunakan *cosine similarity* untuk meningkatkan akurasi *query* pada *web service repository*. *Undergraduate Thesis, Universitas Islam Negeri Maulana Malik Ibrahim*. <http://etheses.uin-malang.ac.id/id/eprint/13421>
- Gao, X. (2013). Towards the Next Generation Intelligent BPM – In the Era of Big Data. In F. Daniel, J. Wang, & B. Weber (Eds.), *Business Process Management* (Vol. 8094, pp. 4–9). https://doi.org/10.1007/978-3-642-40176-3_2

- Gulia, P., & Palak, P. (2017). Component Based Software Development Life Cycle Models: A Comparative Review. *Oriental Journal of Computer Science and Technology*, 10(2), 467–473. <https://doi.org/10.13005/ojcs/10.02.30>
- Hammer, M. (2015). What is Business Process Management? In J. vom Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 1* (pp. 3–16). https://doi.org/10.1007/978-3-642-45100-3_1
- Hepp, M., Leymann, F., Domingue, J., Wahler, A., & Fensel, D. (2005). Semantic business process management: A vision towards using semantic Web services for business process management. *IEEE International Conference on E-Business Engineering (ICEBE'05)*, 535–540. <https://doi.org/10.1109/ICEBE.2005.110>
- Khan, A. I., & Khan, U. A. (2012). An Improved Model for Component Based Software Development. *Software Engineering* 2162-8408, 2, 10.
- Kim, D., Seo, D., Cho, S., & Kang, P. (2019). Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Information Sciences*, 477, 15–29. <https://doi.org/10.1016/j.ins.2018.10.006>
- Lederer, M., Knapp, J., & Schott, P. (2017). The digital future has many names—How business process management drives the digital transformation. *2017 6th International Conference on Industrial Technology and Management (ICITM)*, 22–26. <https://doi.org/10.1109/ICITM.2017.7917889>
- Miller, E. (2005). An Introduction to the Resource Description Framework. *Journal of Library Administration*. <https://doi.org/10.1002/bult.105>

- Misra, S. (2009). Weyuker's Properties, Language Independency and Object Oriented Metrics. In O. Gervasi, D. Taniar, B. Murgante, A. Laganà, Y. Mun, & M. L. Gavrilova (Eds.), *Computational Science and Its Applications – ICCSA 2009* (Vol. 5593, pp. 70–81). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02457-3_6
- Owen, M., & Raj, J. (2003). BPMN and Business Process Management: Introduction to the New Business Process Modeling Standard. *Popkin Software*.
- Shi, Y., & Lu, X. (2010). The Role of Business Intelligence in Business Performance Management. *2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering*, 184–186. <https://doi.org/10.1109/ICIII.2010.522>
- Shomad Bukhori, A. (2020). Analisis dan Perancangan *Software* BPM Sistem. *ILKOMNIKA : Journal of Computer Science and Applied Informatics*.
- Teniente, E., & Weidlich, M. (Eds.). (2018). *Business Process Management Workshops: BPM 2017 International Workshops, Barcelona, Spain, September 10-11, 2017, Revised Papers* (Vol. 308). <https://doi.org/10.1007/978-3-319-74030-0>
- Van Looy, A. (2018). On the Synergies Between Business Process Management and Digital Innovation. In M. Weske, M. Montali, I. Weber, & J. vom Brocke (Eds.), *Business Process Management* (Vol. 11080, pp. 359–375). Springer International Publishing. https://doi.org/10.1007/978-3-319-98648-7_21

- Wang, N., Zeng, N. N., & Zhu, W. (2010). Sensitivity, Specificity, Accuracy, Associated Confidence Interval And ROC Analysis With Practical SAS Implementations. *NorthEast SAS Users Group, Health Care and Life Sciences*.
- Wil M.P. van der Aalst. (2003). Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language. *Queensland University of Technology*.
- Xiao Ying, Chen Deren, & Chen Min. (2004). Research of web services workflow and its key technology based on XPDL. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, 3, 2137–2142. <https://doi.org/10.1109/ICSMC.2004.1400643>
- Yaqin, M., Sarno, R., Rochimah, S. (2020). Measuring Scalable Business Process Model Complexity Based on Basic Control Structure. *International Journal of Intelligent Engineering and Systems*, 13(6), 52–65. <https://doi.org/10.22266/ijies2020.1231.06>